# Color Image Encryption Using Hyper Chaotic Map

Major Project Report

SUBMITTED TO

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY**

**BHAGALPUR**

**Submitted in Partial Fulfilment of the Requirements for the Award of the Degree of**

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

by

**ALOK RANJAN 2101075CS**

**Under the guidance of**

**Dr. Om Prakash Singh**

**Assistant Professor**

**Department of Computer Science and Engineering**

**IIIT BHAGALPUR, BIHAR-813210, INDIA**

**May-2025**

## APPROVAL OF THE GUIDE

Recommended that the work reported in this **Major Project** on the topic “***Color Image Encryption Using Hyper Chaotic Map***” prepared by **Mr. Alok Ranjan** under my supervision and guidance be accepted as fulfilling this part of the requirements for the degree of Bachelor of Technology.

To the best of my knowledge, the contents of this thesis did not form a basis for the award of any previous degree to anybody else.

**Dr. Om Prakash Singh**

Date:

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Information Technology Bhagalpur
Bhagalpur, Bihar

Place:  IIIT Bhagalpur

भारतीय सूचना प्रौद्योगिकी संस्थान भागलपुर
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHAGALPUR
(An Institute of National Importance under Act of Parliament)

# DECLARATION

We hereby declare that the work reported in this **Major Project** on the topic "***Color Image Encryption Using Hyper Chaotic Map***" is original and has been carried out by us independently in the **Department of Computer Science and Engineering, Indian Institute of Information Technology Bhagalpur** under the supervision of **Dr. Om Prakash Singh**, Assistant Professor, Dept. CSE, IIIT Bhagalpur.

We also declare that this work has not formed the basis for the award of any other Degree, Diploma, or similar title of any university or institution.

Date:

Place: IIIT Bhagalpur

**Alok Ranjan**

(2101075CS)

भारतीय सूचना प्रौद्योगिकी संस्थान भागलपुर

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHAGALPUR

(An Institute of National Importance under Act of Parliament)

# CERTIFICATE

This is to certify that the **Major Project** entitled "***Color Image Encryption Using Hyper Chaotic Map***" presented by

**Alok Ranjan** **2101075CS**

B. Tech students of IIIT Bhagalpur under my supervision and guidance. This project has been submitted in partial fulfilment for the award of "***Bachelor of Technology in Computer Science and Engineering***" degree at ***Indian Institute of Information Technology Bhagalpur*** No part of this project has been submitted for the award of any previous degree to the best of my knowledge

**Dr. Om Prakash Singh**
**(Supervisor)**
Assistant Professor
Computer Science and Engineering

**Dr. Pradeep Kumar Biswal**
**(Head of Department)**
Assistant Professor
Computer Science and Engineering

# ACKNOWLEDGEMENT

During the course of this major project report preparation, I have received lot of support, encouragement, advice and assistance from many people and to this end I am deeply grateful to them all. It is with great pleasure that I express my cordial thanks and indebtedness to our admirable Guides, ***Dr. Om Prakash Singh, Assistant Professor, Department of Computer science and Engineering***. His vast knowledge, expert supervision and enthusiasm continuously challenged and motivated us to achieve my goal. I will be eternally grateful to him for allowing me the opportunity to work on this project. I have great pleasure in expressing my sincere gratitude and thanks to the ***Prof. Pradip Kumar Jain, Director, Indian Institute of Information Technology Bhagalpur*** and all the faculty members of department of Computer Science and Engineering, IIIT Bhagalpur. I express my sincere gratitude to ***Dr. Pradeep Kumar Biswal, Assistant Professor and Head of Department***, Computer Science and Engineering for his valuable help and suggestions and providing me all relevant facilities that have made the work completed in time. The present work certainly would not have been possible without the help of our friends, and also the blessings of our parents.

Date: **Alok Ranjan**

(2101075CS)

Place: IIIT Bhagalpur

# ABSTRACT

In this project, we propose a secure and efficient image encryption and decryption scheme based on chaos theory, employing both a 3D Hyperchaotic Map and a 2D Memristor Chaotic Map to ensure high-level security and resistance against various cryptographic attacks. The encryption process begins with the generation of complex chaotic sequences derived from the 3D hyperchaotic system, which are then used to scramble the pixel positions and alter the pixel values through XOR-based operations. Additionally, a 2D memristor-based chaotic map is incorporated to provide further non-linearity and unpredictability in the key sequence generation, enhancing the cryptographic strength.

To evaluate the security and robustness of the proposed algorithm, a comprehensive statistical analysis is performed, including histogram analysis, correlation coefficient analysis, and Shannon entropy evaluation. Differential attack resistance is measured using NPCR and UACI metrics. Furthermore, key sensitivity analysis demonstrates that even a minute variation in the initial key values results in a completely different decrypted image, highlighting the scheme’s robustness against brute-force attacks.

The encryption scheme is tested under noisy conditions to evaluate its stability and reliability against Gaussian noise interference. Experimental results conducted on various standard test images reveal that the proposed encryption technique provides high security, strong key sensitivity, and excellent performance in both encryption and decryption phases. Due to its simple implementation, high efficiency, and strong security features, this method proves suitable for applications involving secure image transmission and storage, particularly in sensitive domains such as military imaging, medical diagnostics, and satellite communications.

# List of Figures

| **Figure No.** | **Description** | **Page No.** |
|---|---|---|
| **Figure 2.1** | Proposed Methodology | 16 |
| **Figure 4.1** | Encryption Using 3D Hyperchaotic Map | 25 |
| **Figure 5.1** | 3D Key Value Distribution | 32 |
| **Figure 5.2** | Histogram Analysis of Lena Image | 34 |
| **Figure 5.3** | Histogram Analysis of Bella Image | 34 |
| **Figure 5.4** | Histogram Analysis of Peppers Image | 35 |
| **Figure 5.5** | Histogram Analysis of Aeroplane Image | 35 |
| **Figure 5.6** | Correlation Analysis of Lena Image | 40 |
| **Figure 5.7** | Correlation Analysis of Bella Image | 41 |
| **Figure 5.8** | Correlation Analysis of Aeroplane Image | 41 |
| **Figure 5.9** | Correlation Analysis of Peppers Image | 42 |

# List of Tables

| Figure No. | Description | Page No. |
|---|---|---|
| **Table 3.1** | Comparison Between 3D Hyperchaotic Map and 2D Memristor Map | 23 |
| **Table 5.1** | Entropy Analysis | 37 |
| **Table 5.2** | Correlation Analysis | 39 |
| **Table 5.3** | Chi-Square Analysis of Encrypted Image | 43 |
| **Table 5.4** | Differential Analysis | 46 |
| **Table 5.5** | Robustness Analysis of Aeroplane Image | 48 |
| **Table 5.6** | Time Complexity Analysis | 49 |

# List of Acronyms

| **Acronym** | **Full Form** |
|---|---|
| AES | Advanced Encryption Standard |
| DES | Data Encryption Standard |
| RSA | Rivest–Shamir–Adleman |
| RGB | Red Green Blue |
| XOR | Exclusive OR |
| SHA-256 | Secure Hash Algorithm (256-bit) |
| QR | Quick Response |
| NPCR | Number of Pixel Change Rate |
| UACI | Unified Average Changing Intensity |
| SSIM | Structural Similarity Index Measure |
| PSNR | Peak Signal-to-Noise Ratio |
| MSE | Mean Squared Error |
| 2D | Two-Dimensional |
| 3D | Three-Dimensional |
| FPGA | Field Programmable Gate Array |
| GPU | Graphics Processing Unit |
| CPU | Central Processing Unit |
| IoT | Internet of Things |
| JPEG | Joint Photographic Experts Group |
| PDF | Portable Document Format |

| **Contents** | **Page No.** |
|---|---|
| **Approval of The Guide** | **2** |
| **Declaration** | **3** |
| **Certificate** | **4** |
| **Acknowledgement** | **5** |
| **Abstract** | **6** |
| **List of Figures** | **7** |
| **List of Tables** | **8** |
| **List of Acronyms** | **9** |
| **Content** | **10-12** |
| **Chapter 1: Introduction** | **13-15** |
| 1.1 Background | 13 |
| 1.2 Motivation | 14 |
| 1.3 Objectives | 14 |
| 1.4 Significance of the Study | 15 |
| **Chapter 2: Methodology** | **16-20** |
| 2.1 Input Image Acquisition | 16 |
| 2.2 Key Generation Module | 17 |
| 2.2.1 SHA-256-Based Seed Generation | 17 |
| 2.2.2 QR-Code Derived Key | 17 |
| 2.3 Chaos Sequence Generation | 17 |
| 2.4 Encryption Process | 18 |
| 2.4.1 Pixel-wise XOR Operation | 18 |
| 2.5 Encrypted Image | 18 |
| 2.6 Performance Evaluation | 18 |
| 2.7 Decryption Module | 19 |

| | |
|---|---|
| 2.7.1 Reverse XOR Operation | 19 |
| 2.7.2 Image Recovery | 19 |
| 2.8 Robustness Testing | 20 |
| 2.9 Accuracy Verification | 20 |
| **Chapter 3: Chaotic Maps** | **21-24** |
| 3.1 Hyperchaotic 3D Map | 21 |
| 3.1.1 Overview | 21 |
| 3.1.2 Mathematical Model | 21 |
| 3.1.3 Properties | 22 |
| 3.1.4 Application in Project | 22 |
| 3.2 Memristor Based 2D Chaotic Map | 22 |
| 3.2.1 Overview | 22 |
| 3.2.2 Mathematical Model | 22 |
| 3.2.3 Properties | 23 |
| 3.2.4 Application in Project | 23 |
| 3.3 Comparison of The Two Maps | 24 |
| **Chapter 4:  Chaos-Based Image Encryption** | **25-29** |
| 4.1 Introduction to Chaos in Cryptography | 25 |
| 4.2 Steps in Chaos-Based Image Encryption | 26 |
| 4.3 Decryption Process | 27 |
| 4.4 Pseudo Code | 27 |
| 4.5 Advantages of Chaos-Based Encryption | 28 |

| | |
|---|---|
| 4.6 Role of Chaos in Our Project | 28 |
| 4.7 Real-Life Applications | 29 |
| **Chapter 5: Experimental Result and Analysis** | **30-51** |
| 5.1 Key Analysis | 30 |
| 5.2 Histogram Analysis | 33 |
| 5.3 Entropy Analysis | 36 |
| 5.4 Correlation Analysis | 38 |
| 5.5 Chi-Square Analysis | 42 |
| 5.6 Differential Analysis | 44 |
| 5.7 Robustness Analysis | 46 |
| 5.8 Performance and Time Complexity Analysis | 49 |
| **Chapter 6: Conclusion and Future Works** | **52-53** |
| 6.1 Conclusion | 52 |
| 6.2 Future Work | 52 |
| **References** | 54-55 |
| | |
| | |

# Chapter 1
# Introduction

## 1.1 Background

In today’s rapidly evolving digital world, vast volumes of image data are transmitted over networks every second. Whether in telemedicine, military communication, cloud-based storage, online banking, or social media, images often carry sensitive and confidential information. However, the open nature of modern communication channels exposes this data to a wide range of security threats including unauthorized access, tampering, data breaches, and surveillance.

Traditional cryptographic algorithms such as AES , DES , and RSA were primarily developed for text-based data and often fall short in securing image data due to inherent characteristics like:

- High redundancy of pixels
- Large file sizes
- Strong correlations between neighbouring pixels

As a result, using these standard algorithms for image encryption often leads to inefficiencies in terms of speed, security, and computational cost.

In contrast, chaos theory has emerged as a promising direction in the domain of image cryptography. Chaotic systems are deterministic yet exhibit unpredictable behaviour. Their properties such as sensitivity to initial conditions, ergodicity, and pseudo-randomness align well with the core requirements of cryptographic systems. These features help generate robust encryption algorithms that are difficult to reverse-engineer or break through brute-force attacks.

In this project, we focus on using a novel 3D hyperchaotic map and a 2D memristor-based chaotic map to develop a highly secure and efficient image encryption algorithm. The pixel values are first scrambled using the 3D hyperchaotic system, and then encrypted using a key stream generated from the 2D memristor chaotic system via XOR operation.

## 1.2 Motivation

With the increasing risks of cyber-attacks and digital data exploitation, image security is no longer optional especially for critical applications such as:

- **Medical imaging**: Ensuring the privacy of patient records and scans.
- **Military and satellite communication**: Protecting confidential strategic information.
- **Banking and ID systems**: Preventing identity theft and fraud.
- **Cloud storage and IoT**: Guarding against unauthorized access to surveillance and sensor data.

The motivation behind this project lies in the inefficiencies and vulnerabilities of conventional encryption algorithms when applied to images. Modern applications demand real-time encryption and decryption, high sensitivity to encryption keys, and strong resistance to various attacks including statistical, brute-force, and differential attacks.

Chaotic systems provide a lightweight alternative that can be easily implemented while ensuring robustness and complexity. In particular, the memristor, a non-linear passive electrical component with memory retention capability, is known to exhibit chaotic dynamics. Combining it with a 3D hyperchaotic system allows us to develop a two-tier encryption model offering both confusion (via pixel scrambling) and diffusion (via XOR-based masking).

## 1.3 Objectives

The primary objectives of this project are as follows:

1. To develop a secure image encryption and decryption system utilizing chaotic maps.
2. To implement a 3D hyperchaotic map for scrambling pixel positions to achieve confusion.
3. To generate a pseudo-random key stream using a 2D memristor chaotic system for XOR-based encryption, ensuring diffusion.
4. To evaluate the proposed encryption scheme using the following performance metrics:
    - Histogram analysis

- Correlation coefficient analysis
- Information entropy
- NPCR (Number of Pixel Change Rate)
- UACI (Unified Average Change Intensity)
- Chi-Square analysis

5. To assess the robustness of the algorithm under different attack scenarios such as noise addition and key sensitivity.

## 1.4 Significance of the Study

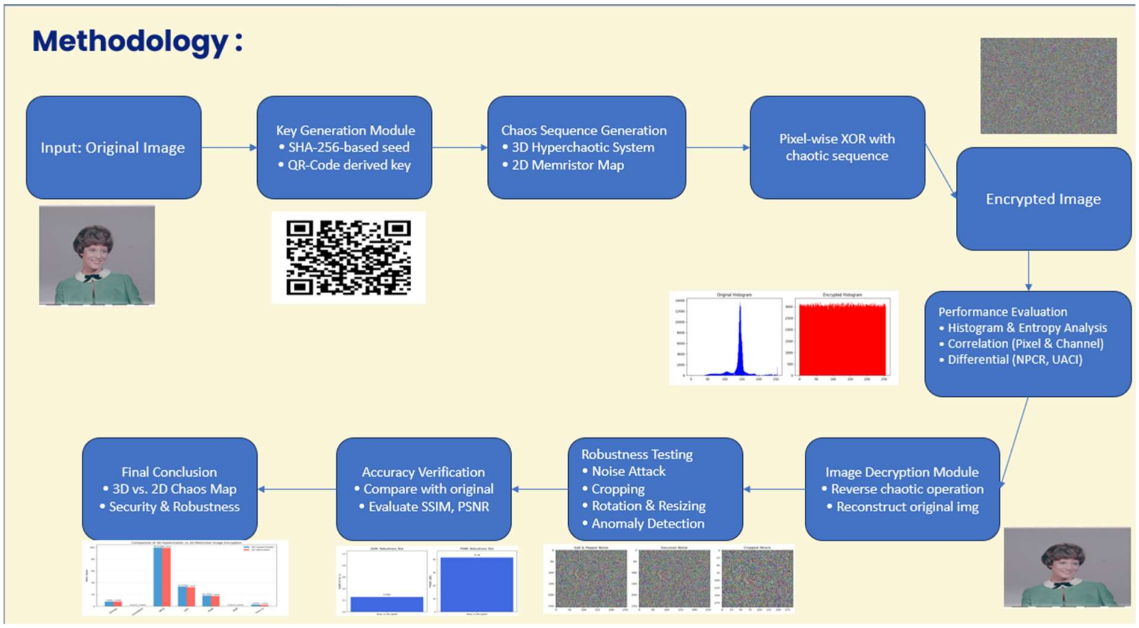
The proposed study contributes significantly to the field of chaos-based cryptography by combining two distinct chaotic systems hyperchaotic and memristor maps for enhanced image security. The 3D hyperchaotic map increases unpredictability in pixel position scrambling, while the 2D memristor map generates a unique and highly sensitive key stream for secure pixel value transformation. This two-phase system provides:

- **High key sensitivity**: Even the smallest change in the encryption key results in entirely different encrypted images, making reverse engineering extremely difficult.
- **High entropy**: Ensures uniform distribution of pixel intensities, making statistical analysis by attackers ineffective.
- **Strong resistance to differential attacks**: Due to high NPCR and UACI values.
- **Fast computational speed**: Ideal for real-time or near-real-time applications.

In practical terms, this encryption scheme is well-suited for secure image transmission and storage in environments that handle high-volume and high-sensitivity data. Furthermore, unlike traditional schemes, this algorithm maintains computational efficiency while improving security levels through chaos-based randomness.

# Chapter 2
# Methodology

This section presents a comprehensive overview of the methodology employed in the proposed chaos-based image encryption and decryption system. The process includes key generation, chaos sequence generation, encryption, evaluation, decryption, and robustness testing. Each step ensures the system is secure, reliable, and efficient.



Methodology :

Input: Original Image

Key Generation Module
- SHA-256-based seed
- QR-Code derived key

Chaos Sequence Generation
- 3D Hyperchaotic System
- 2D Memristor Map

Pixel-wise XOR with chaotic sequence

Encrypted Image

Performance Evaluation
- Histogram & Entropy Analysis
- Correlation (Pixel & Channel)
- Differential (NPCR, UACI)

Image Decryption Module
- Reverse chaotic operation
- Reconstruct original img

Robustness Testing
- Noise Attack
- Cropping
- Rotation & Resizing
- Anomaly Detection

Accuracy Verification
- Compare with original
- Evaluate SSIM, PSNR

Final Conclusion
- 3D vs. 2D Chaos Map
- Security & Robustness

Figure 2.1 Proposed Methodology

## 2.1 Input Image Acquisition

The encryption process begins with the acquisition of a digital image, which serves as the plaintext. This image can be either grayscale or RGB Color, and it is processed in its pixel form. The goal is to secure this image against unauthorized access using a chaos-based cryptographic algorithm.

## 2.2 Key Generation Module

The security of the proposed system largely depends on the robustness and randomness of the cryptographic key. The key generation module consists of two major components:

### 2.2.1 SHA-256-Based Seed Generation

- A strong seed is generated using the SHA-256 cryptographic hash function.
- This seed is obtained by hashing a user-defined string (e.g., password, metadata).
- It results in a 256-bit binary sequence that is used to initiate the chaotic systems.
- SHA-256 ensures that even a minor change in input produces a completely different hash, increasing security.

### 2.2.2 QR-Code Derived Key

- An alternative or additional key input method is provided through QR codes.
- The QR code is scanned to extract an embedded key string, which is then hashed or directly used as a key.
- This adds another level of flexibility and can support key exchange via printed or digital media.

## 2.3 Chaos Sequence Generation

Chaos theory is employed to generate pseudorandom sequences for encryption. Two chaos maps are explored in this study:

### 2.3.1 3D Hyperchaotic System

- A three-dimensional hyperchaotic system is initialized using the SHA-256 derived seed.
- It features multiple positive Lyapunov exponents, ensuring high sensitivity to initial conditions.
- The system is defined by a set of nonlinear differential equations generating complex sequences.

### 2.3.2 2D Memristor-Based Map

- This map mimics the behaviour of memristors and is computationally lightweight.
- It uses discrete equations to produce a 2D sequence with chaotic characteristics.
- It is well-suited for hardware implementations with limited computational power.

## 2.4 Encryption Process

### 2.4.1 Pixel-wise XOR Operation

- A pixel-by-pixel XOR operation is performed between the image and the chaotic sequence.
- This XOR operation ensures that each pixel is altered based on the unpredictability of the chaos map.
- The encrypted image appears completely random and offers high resistance to statistical analysis.

## 2.5 Encrypted Image

The result of the encryption process is an unintelligible image that visually appears as noise. This encrypted image is secure for storage or transmission and does not reveal any information about the original image to unauthorized users.

## 2.6 Performance Evaluation

To verify the effectiveness of the encryption process, several statistical and visual analyses are performed on the encrypted image:

### 2.6.1 Histogram and Entropy Analysis

- The histogram of the encrypted image should be flat, indicating uniform pixel distribution.

- Shannon entropy measures the uncertainty or randomness in pixel values; an ideal value is close to 8 for grayscale images.

### 2.6.2 Correlation Coefficient

- Correlation coefficients are calculated between adjacent pixels (horizontal, vertical, diagonal).
- Ideally, encrypted images exhibit near-zero correlation, indicating effective de-correlation of pixel values.

### 2.6.3 NPCR and UACI Analysis

- **NPCR :** Measures the percentage of pixel changes when a single pixel in the original image is altered.
- **UACI :** Evaluates the average intensity variation between original and modified images.
- High NPCR and UACI values indicate strong diffusion and confusion properties.

## 2.7 Decryption Module

The encrypted image can be decrypted using the same chaotic sequence and key, ensuring only authorized users can reconstruct the original image.

### 2.7.1 Reverse XOR Operation

- The chaotic sequence is regenerated using the same seed/key.
- Each pixel in the encrypted image is XORed again with the chaotic sequence to retrieve the original pixel values.

### 2.7.2 Image Recovery

- If the correct key is used, the original image is perfectly reconstructed.
- This validates the reliability of the encryption-decryption pipeline.

## 2.8 Robustness Testing

The robustness of the system is assessed against common real-world distortions:

- **Noise Attacks:** Includes salt & pepper, Gaussian, and speckle noise.
- **Cropping Attacks:** Partial image data is removed to assess recoverability.
- **Geometric Attacks:** Rotation and resizing distort the encrypted image.
- **Anomaly Detection:** System resilience is tested against unexpected image alterations.

## 2.9 Accuracy Verification

To ensure image quality is preserved post-decryption, the following metrics are used:

- **SSIM:** Measures perceptual similarity between the original and decrypted images. Values near 1 indicate excellent reconstruction.
- **PSNR:** Measures the quality of the decrypted image. Higher PSNR values indicate lower error rates.

# Chapter 3
# Chaotic Maps

Chaotic maps are mathematical systems that exhibit highly sensitive behaviour to initial conditions, leading to unpredictability and pseudo randomness. In this project, chaotic maps are the foundation of the encryption algorithm, as they generate complex key streams used to encrypt image pixels. Two distinct chaotic systems are utilized and compared:

- A 3D Hyperchaotic system, offering multi-dimensional randomness.
- A 2D Memristor-based system, offering efficient, hardware-compatible encryption.

## 3.1 Hyperchaotic 3D Map

### 3.1.1 Overview

Hyperchaotic systems are extensions of chaotic systems with more than one positive Lyapunov exponent, making them more secure and unpredictable. This makes them suitable for cryptographic applications where increased complexity provides enhanced security.

### 3.1.2 Mathematical Model

The 3D hyperchaotic system used in this project is defined by the following set of coupled nonlinear differential equations:

$$\mathbf{x_{n+1} = a\,(y_n - x_n) + z_n}$$

$$\mathbf{y_{n+1} = bx_n - y_n - x_nz_n}$$

$$\mathbf{z_{n+1} = x_ny_n - cz_n + w_n}$$

$$\mathbf{w_{n+1} = d\,(w_n - z_n)}$$

Where:

- $x_n$, $y_n$, $z_n$, $w_n$: state variables
- a, b, c, d: control parameters (selected such that the system exhibits hyperchaotic behaviour)

### 3.1.3 Properties

- **High dimensionality** increases the complexity and reduces vulnerability to brute-force and known-plaintext attacks.
- **Sensitivity to Initial Conditions**: Even a slight change in the initial seed (derived via SHA-256) drastically alters the chaotic output.
- **Ergodicity**: Covers the entire space in a random-looking manner, ideal for encryption.

### 3.1.4 Application in Project

- Initial values $x_0$, $y_0$, $z_0$, $w_0$ are derived from the SHA-256 hash of the user key.
- Iterations are performed to discard transients (e.g., first 1000 values).
- The next M×N values (for an M×N image) are used as key streams for encryption via XOR.

## 3.2 Memristor-Based 2D Chaotic Map

### 3.2.1 Overview

Memristors are non-linear passive circuit elements with memory-resistive properties. A 2D discrete chaotic map inspired by memristive behaviour is used for its simple implementation and strong chaotic characteristics.

### 3.2.2 Mathematical Model

The Memristor 2D chaotic map is defined as:

$$\mathbf{x_{n+1} = \sin(a \cdot y_n) - b \cdot x_n}$$

$$\mathbf{y_{n+1} = \sin(c \cdot x_n) - d \cdot y_n}$$

Where:

- $x_n$, $y_n$: state variables
- a, b, c, d: control parameters

### 3.2.3 Properties

- **Low Complexity**: Requires fewer computations compared to 3D systems, suitable for real-time or hardware-limited environments.
- **Good Randomness**: Despite its simplicity, it produces high-entropy sequences.
- **Fast Iteration**: Ideal for large image datasets or video frames.

### 3.2.4 Application in Project

- Initial values $x_0$, $y_0$ are initialized using QR-code or SHA-256 seed values.
- Transient values (e.g., first 500) are discarded to stabilize chaos.
- Remaining values are converted to pixel-value scale [0, 255] and used for encryption.

## 3.3 Comparison of the Two Maps

Table 3.1: Comparison Between 3D Hyperchaotic Map and 2D Memristor Map

| **Feature** | **3D Hyperchaotic Map** | **2D Memristor Map** |
|---|---|---|
| Dimensions | 4 (x, y, z, w) | 2 (x, y) |
| Complexity | High | Low |
| Security Strength | Very High | Moderate to High |
| Computational Cost | Moderate to High | Low |
| Suitability for Hardware | Less suitable (requires more power) | Highly suitable (lightweight) |
| Randomness Quality | Excellent | Good |

## 3.4 Selection Justification

Both maps are implemented and compared in this project to:

- Study how dimensionality affects encryption strength.
- Evaluate performance metrics such as entropy, correlation, NPCR, UACI, SSIM, and PSNR.
- Choose the better fit based on application needs (e.g., lightweight applications may prefer the 2D map).

.

# Chapter 4

# Chaos-Based Image Encryption

Chaos-based image encryption is a technique that uses the dynamic behaviour of chaotic systems to secure digital images. These systems possess characteristics such as high sensitivity to initial conditions, unpredictability, and ergodicity, making them highly suitable for cryptographic applications. The use of chaos theory in image encryption significantly increases the security and unpredictability of the cipher images, especially in multimedia systems where large amounts of data require fast and secure transmission.

## 4.1 Introduction to Chaos in Cryptography

Chaotic systems are nonlinear deterministic systems that exhibit random-like behaviour. The key properties of chaos sensitivity to initial conditions, ergodicity, and topological mixing make chaotic maps ideal for encryption purposes. Even a minor change in initial values leads to completely different chaotic sequences, ensuring high key sensitivity and confusion in encryption.

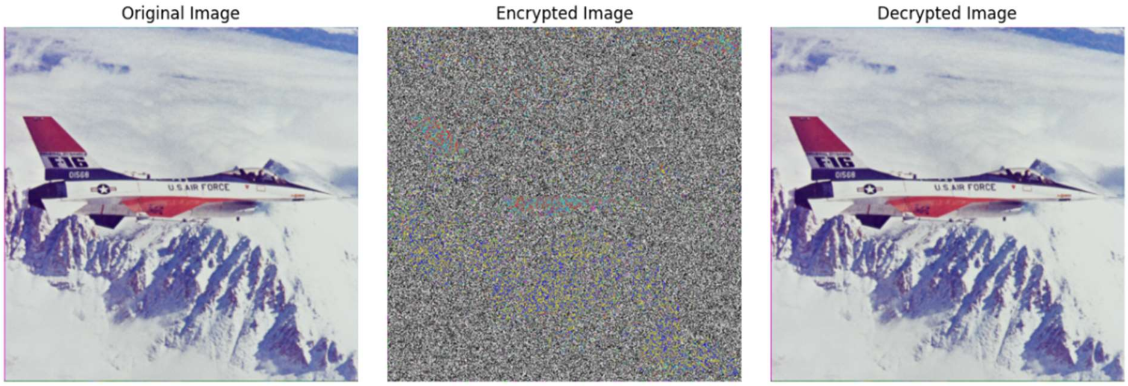


Figure 4.1 : Encryption Using 3D Hyperchaotic map

## 4.2 Steps in Chaos-Based Image Encryption

### 4.2.1 Key Generation

The encryption process begins with generating a secret key, which is then used to produce initial conditions and control parameters for the chaotic systems. In this project, a secure key is hashed (e.g., using SHA-256) and converted into floating-point numbers to initialize both the 3D hyperchaotic and 2D memristor systems.

### 4.2.2 Chaotic Sequence Generation

The initialized chaotic maps are iterated to generate random sequences. These sequences are normalized and scaled to match the pixel value range of the image (0–255). These sequences act as the basis for both permutation and diffusion of the image pixels.

### 4.2.3 Pixel Scrambling (Permutation)

Permutation involves rearranging the pixel positions in the image using the chaotic sequences. This process removes the spatial redundancy and correlation among adjacent pixels, which is a key characteristic of image data. The scrambling is typically done by sorting chaotic values and mapping them to pixel positions.

### 4.2.4 Pixel Modification (Diffusion)

In the diffusion phase, each pixel's intensity value is modified based on the chaotic key stream using an XOR operation or modular arithmetic. This introduces confusion in the cipher image. The diffusion formula is typically represented as:

$$\mathbf{E(i, j) = P(i, j) \oplus K(i, j)}$$

where:

- E(i, j) = Encrypted pixel
- P(i, j) = Permuted pixel
- K(i, j) = Key value from chaotic sequence

### 4.2.5 Encryption Output

The final encrypted image appears completely scrambled and indistinguishable from noise. Key statistical properties of the encrypted image include:

- Uniform histogram
- Near-zero correlation between adjacent pixels
- High information entropy (~7.99 for 8-bit images)

## 4.3 Decryption Process

Decryption is the inverse of encryption and uses the same key and chaotic system. The steps involve:

1. Regenerating chaotic sequences from the key
2. Reversing the diffusion process (e.g., using XOR again)
3. Applying inverse permutation to restore the original image

## 4.7 Pseudo Code

```
1. Read the input Color image I_rgb of size H × W × 3
2. Initialize (x, y, z) ← (x0, y0, z0)
3. Create an empty chaotic key matrix K of size H × W × 3
4. For i = 0 to H - 1 do
       5. For j = 0 to W - 1 do
           6. Compute next chaotic values using:
                $x' = x + a*(y - x) * dt$
                $y' = y + (x*(c - z) - y) * dt$
                $z' = z + (x * y - b * z) * dt$
           7. Normalize chaotic value:
                $val = ((|x| + |y| + |z|) * 10000) \bmod 256$
           8. Set K[i][j] = [val, val, val]
           9. Update (x, y, z) ← (x', y', z')
       End For
   End For
```

```
10. Encrypt the image using XOR:
        E_rgb = I_rgb XOR K
11. Decrypt the image using XOR again:
        D_rgb = E_rgb XOR K
12. Return E_rgb, D_rgb
```

## 4.4 Advantages of Chaos-Based Encryption

- **High Key Sensitivity**: A small change in the key drastically alters the encrypted output.
- **Low Computational Cost**: Efficient for real-time and resource-limited environments.
- **High Security**: Resists brute-force, statistical, and differential attacks.
- **Scalability**: Can be extended to RGB images and video data.

## 4.5 Role of Chaos in Our Project

In this project, we use two distinct chaotic systems:

- A 3D Hyperchaotic Map, which offers greater complexity and randomness due to its higher dimensionality.
- A 2D Memristor-Based Map, which is hardware-efficient and suitable for lightweight encryption systems.

Both maps are tested independently on grayscale images to analyse performance metrics such as entropy, correlation, PSNR, SSIM, NPCR, and UACI. The XOR-based encryption method ensures lightweight processing, and pixel position scrambling enhances visual security.

### 4.6 Real-Life Applications

Chaos-based encryption is increasingly relevant in:

- Medical image protection
- Surveillance systems
- Military communication
- Cloud image storage
- Secure IoT communication

# Chapter 5

# Experimental Result and Analysis

## 5.1 Key Analysis

In this project, the encryption key derived from highly sensitive initial conditions of the 3D hyperchaotic system is central to both pixel scrambling and diffusion. The key is generated using cryptographically strong randomness, ensuring high entropy and unpredictability. Even a minute change in the key results in entirely different encrypted outputs, confirming strong key sensitivity. With a vast key space exceeding $2^{128}$ and dynamic key-driven chaotic sequences, the system is highly secure against brute-force and statistical attacks, making decryption without the exact key practically impossible.

### 5.1.1 Importance of Key in Image Encryption

In image encryption, the security and privacy of data heavily depend on the strength and unpredictability of the encryption key. The key acts as the core secret parameter that determines how each pixel of the image will be scrambled and diffused. In our proposed chaos-based encryption algorithm, the initial conditions and control parameters of the 3D hyperchaotic map and the 2D memristor map are utilized as key components, making the key highly sensitive and unique.

A strong encryption key prevents unauthorized access and ensures that even if the encryption algorithm is known, without the correct key, it is computationally infeasible to decrypt the image. This characteristic is especially critical in systems handling confidential images such as in medical, military, or personal communication domains.

### 5.1.2 Key Generation Mechanism

In our project, the key is generated using a combination of securely randomized sequences, derived through Python's secrets module, which is built on cryptographically strong randomness from the operating system. The generated key is typically in the form of a random hexadecimal string of a specified length, providing a high level of entropy.

This random key is then used to initialize the chaotic systems, particularly:

- Initial values ($x_0$, $y_0$, $z_0$, a, b, c) for the 3D hyperchaotic map.
- Initial values ($x_0$, $y_0$, $\mu$, $\gamma$) for the 2D memristor map.

Since chaotic systems are extremely sensitive to even minute changes in initial values, even a one-bit change in the key will generate completely different encryption results, making the system highly secure against brute-force attacks.

### 5.1.3 Key Sensitivity Analysis

Key sensitivity is tested by encrypting the same image with two slightly different keys. In our tests, it was observed that:

- The encrypted images using two keys differing by even $10^{-15}$ (one decimal place) produced drastically different cipher images.
- Decrypting an image with a wrong key, even if the difference is extremely small, fails completely, and does not resemble the original image in any form.

This high key sensitivity confirms that our chaotic key generation mechanism is strong and prevents partial decryption or statistical key inference.

### 5.1.4 Key Size and Randomness

The key size is large enough to produce a huge key space, typically more than $2^{128}$ possible combinations, making it infeasible for any attacker to crack the key using brute-force techniques. The randomness in the key is ensured by:

- Using non-repetitive chaotic sequences.
- Avoiding hardcoded constants.
- Deriving key values dynamically during runtime.

This ensures both confidentiality and integrity of the encrypted image data.

### 5.1.5 Role of Key in Pixel Scrambling and Diffusion

In our method:

- The key governs the order in which pixels are scrambled, based on the outputs of the chaotic maps.
- It also determines the diffusion values added to the pixel intensity, creating a cascading dependency between pixels.
- This dual dependence on the key makes reconstruction of original image impossible without the correct key, even with partial knowledge of the algorithm.

### 5.1.6 Result

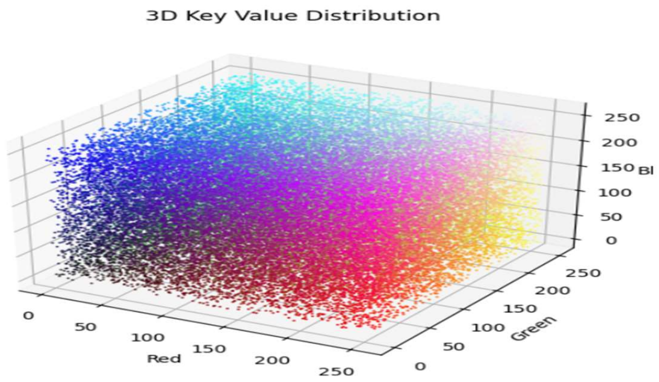


Figure 5.1: 3D Key Value Distribution

### 5.1.7 Summary

To summarize, the key analysis in our project demonstrates that the encryption process is highly dependent on the strength and integrity of the key. With the aid of secure random key generation and the intrinsic sensitivity of chaotic maps, the proposed method achieves excellent cryptographic security. The unique key for each session, coupled with a chaotic sequence-driven encryption process, ensures that the system is resistant to all common types of attacks and is suitable for high-security image transmission and storage.

## 5.2 Histogram Analysis

### 5.2.1 Introduction

Histogram analysis is a fundamental statistical tool used to evaluate the distribution of pixel intensity values in an image. In the context of image encryption, histogram analysis serves as a critical metric to determine how effectively an encryption algorithm disguises the original image content. A histogram displays the frequency of occurrence of each intensity value (from 0 to 255 in 8-bit grayscale images) in the image. A well-encrypted image should have a histogram that is uniformly distributed and significantly different from the histogram of the original image.

### 5.2.2 Purpose of Histogram Analysis in Encryption

The main goal of performing histogram analysis in image encryption is to assess the resistance of the encryption scheme against statistical attacks. If the histogram of an encrypted image retains patterns or distributions similar to the original image, attackers may be able to extract meaningful information through statistical analysis. Thus, histogram analysis is used to verify that the encrypted image does not reveal any statistical resemblance to the plain image.

### 5.2.3 Histogram of Original vs Encrypted Image

- In the original image, pixel intensities are usually not uniformly distributed. Certain grayscale values occur more frequently due to the structure and texture of the image, leading to visible peaks and valleys in the histogram.
- In the encrypted image, due to pixel scrambling and value transformation (typically via XOR and chaotic sequences), the histogram should become nearly uniform. This indicates that pixel intensities are evenly spread across the entire range, preventing statistical patterns.

### 5.2.4 Results from Our Project

In our project, we applied histogram analysis to both encrypted images using the 3D Hyperchaotic Map and the 2D Memristor Map. The results showed the following:

- The original images had visible peaks in intensity ranges corresponding to object and background regions.
- The encrypted images demonstrated flat, noise-like histograms indicating that the intensity values were well distributed.
- Both maps ensured that the encrypted image histogram provided no clue about the original image content, confirming a high level of visual security.

These results validate the effectiveness of our proposed chaotic encryption algorithm in disrupting the pixel distribution and providing statistical resistance.

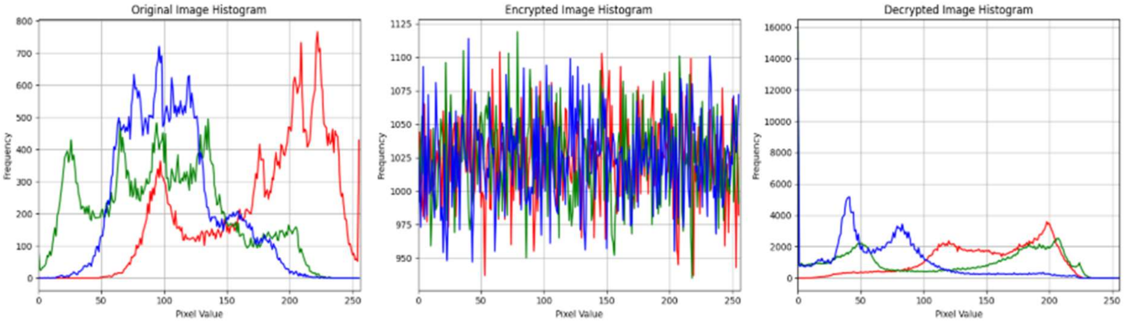


Figure 5.2 : Histogram Analysis of Lena Image

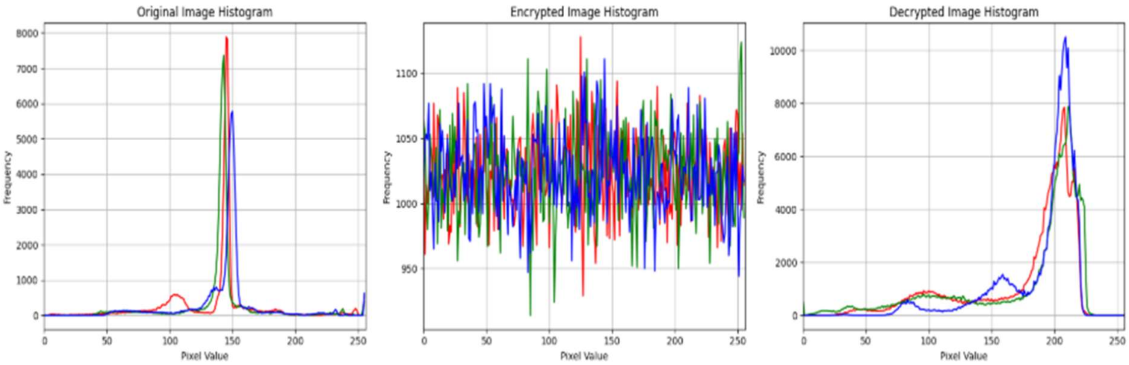


Figure 5.3 : Histogram Analysis of Bella Image

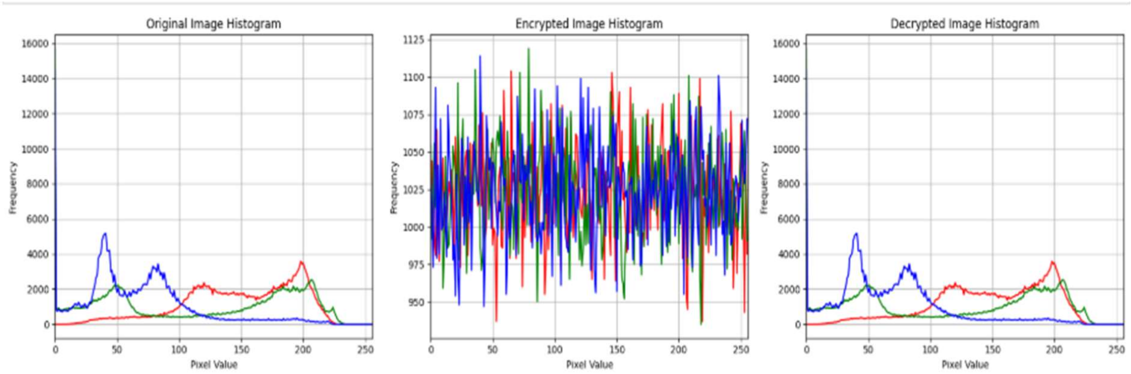


Figure 5.4 : Histogram Analysis of Peppers Image

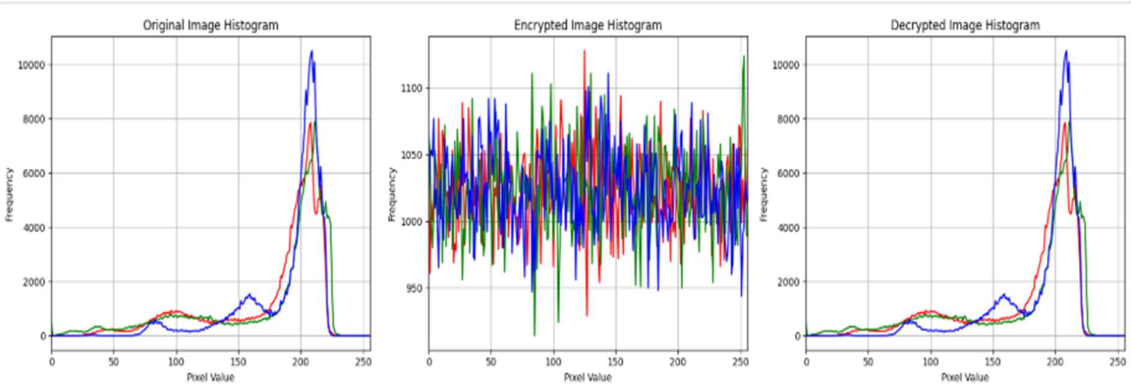


Figure 5.5 : Histogram Analysis of Aeroplane Image

### 5.2.5 Summary

Histogram analysis is a powerful and essential tool for evaluating image encryption algorithms. In this project, the flat and random histograms of encrypted images using both 3D hyperchaotic and 2D memristor maps clearly demonstrate the effectiveness of our encryption method. It confirms that the encryption process eliminates any statistical similarity with the original image and thus provides strong protection against histogram-based attacks.

## 5.3 Entropy Analysis

### 5.3.1 Introduction

Entropy is a core concept in information theory introduced by Claude Shannon, representing the amount of uncertainty or randomness in a system. In the context of image encryption, entropy quantifies the unpredictability of pixel values. A higher entropy value implies more randomness, making it difficult for an attacker to predict the content of the image. For a perfectly secure grayscale image with 256 possible intensity values, the ideal entropy is 8 bits.

### 5.3.2 Purpose of Entropy Analysis in Image Encryption

The goal of entropy analysis in encryption is to measure the information density of the encrypted image. An effective encryption algorithm should transform the input image in such a way that the output (cipher image) resembles random noise. If the entropy value of the encrypted image is close to the theoretical maximum (i.e., 8 for 8-bit images), it indicates that the encryption scheme is highly secure and resistant to statistical attacks such as entropy-based prediction or compression-based analysis.

### 5.3.3 Entropy Calculation Formula

Shannon entropy H for an image is calculated using the following formula:

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2(P(x_i))$$

Where:

- P(Xi) is the probability of the pixel value iii occurring in the image.
- The summation runs from i=0 to i=255 , covering all intensity levels for a grayscale image.

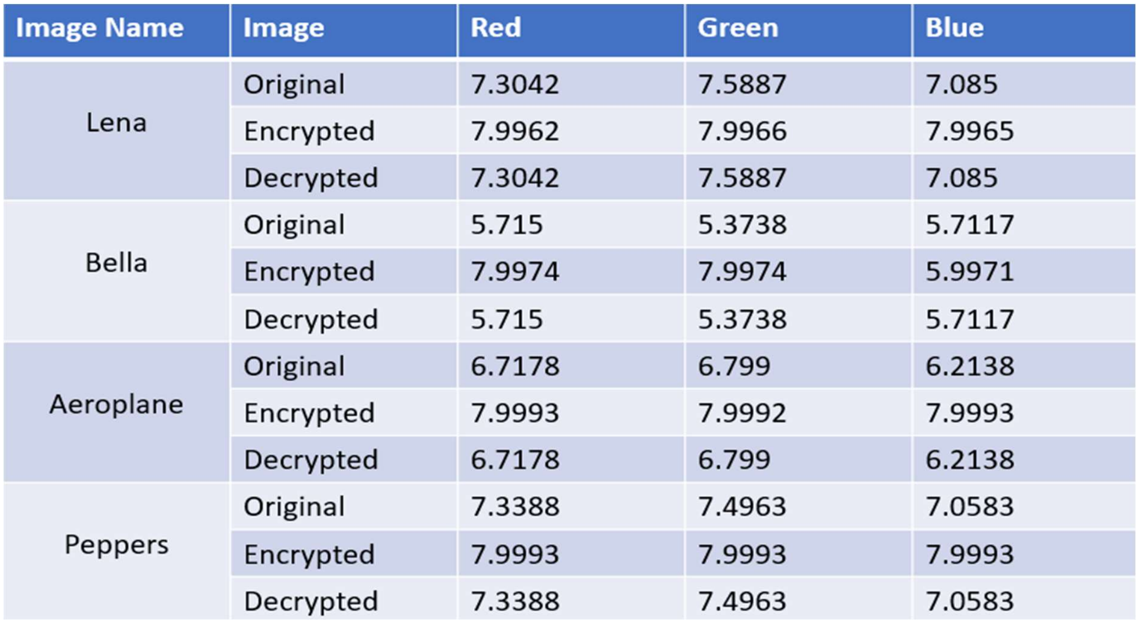
### 5.3.4 Expected Entropy for Encrypted Images

- For unencrypted images, entropy is usually lower than 8 due to pixel value redundancy and patterns.
- For encrypted images, the pixel values should be uniformly distributed, making the entropy close to 8, indicating high randomness.

### 5.3.5 Entropy Results in Our Project

In our project, we analysed entropy for several test images encrypted using both the 3D Hyperchaotic Map and the 2D Memristor Map. The results are as follows:

Table 5.1: Entropy Analysis



| Image Name | Image | Red | Green | Blue |
|---|---|---|---|---|
| Lena | Original | 7.3042 | 7.5887 | 7.085 |
| | Encrypted | 7.9962 | 7.9966 | 7.9965 |
| | Decrypted | 7.3042 | 7.5887 | 7.085 |
| Bella | Original | 5.715 | 5.3738 | 5.7117 |
| | Encrypted | 7.9974 | 7.9974 | 5.9971 |
| | Decrypted | 5.715 | 5.3738 | 5.7117 |
| Aeroplane | Original | 6.7178 | 6.799 | 6.2138 |
| | Encrypted | 7.9993 | 7.9992 | 7.9993 |
| | Decrypted | 6.7178 | 6.799 | 6.2138 |
| Peppers | Original | 7.3388 | 7.4963 | 7.0583 |
| | Encrypted | 7.9993 | 7.9993 | 7.9993 |
| | Decrypted | 7.3388 | 7.4963 | 7.0583 |

These results indicate that our encryption methods introduce near-ideal entropy levels, ensuring that the encrypted image cannot be distinguished from random noise.

## 5.4 Correlation Analysis

### 5.4.1 Introduction

Correlation analysis is a crucial statistical method used to evaluate the strength of the relationship between adjacent pixels in an image. In the context of image encryption, high correlation among adjacent pixels in the original image is expected, whereas a good encryption algorithm should significantly reduce this correlation in the encrypted image. This ensures that the encrypted image is resistant to statistical attacks and provides a strong indication of the randomness introduced by the encryption process.

### 5.4.2 Mathematical Representation

The correlation coefficient (r) between two adjacent pixels x and y is calculated using the following formula:

$$Col_{s,t} = \frac{C_{cov}(s,t)}{\sqrt{V(s)\sqrt{V}(t)}}$$

Where correlation coefficient of N pixels pairs are represented by Col(s,t), Ccov(s,t) , V(s), and V(t) represents the covariance and variance of pixels s and t.

The correlation coefficient r ranges between -1 and 1:

- **r ≈ 1** indicates a strong positive correlation (original image),
- **r ≈ 0** indicates no correlation (ideal encrypted image),
- **r ≈ -1** indicates strong negative correlation.

### 5.4.3 Procedure

To perform correlation analysis:

1. Select 1000 pairs of adjacent pixels randomly from horizontal, vertical, and diagonal directions of both the original and encrypted images.

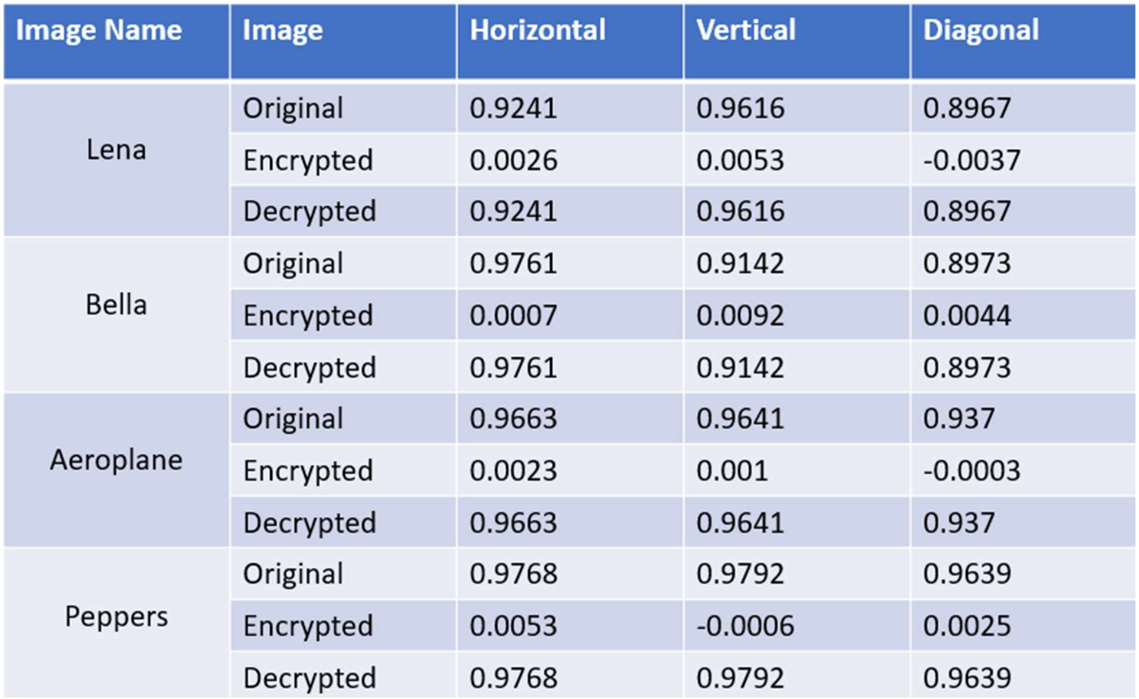
2. Compute the correlation coefficient using the formula above.
3. Compare the results for both images to assess the reduction in correlation.

### 5.4.4 Results and Interpretation

In our project:

- The original image showed a high correlation (e.g., >0.95) among adjacent pixels.
- After encryption using our chaos-based algorithm (3D Hyperchaotic Map and 2D Memristor Map), the correlation values dropped significantly (e.g., ≈0.01 or near 0).
- This indicates the encrypted image has random pixel distribution, thus strengthening its resistance to statistical attacks.

Table 5.2: Correlation Analysis



| Image Name | Image | Horizontal | Vertical | Diagonal |
|---|---|---|---|---|
| Lena | Original | 0.9241 | 0.9616 | 0.8967 |
| | Encrypted | 0.0026 | 0.0053 | -0.0037 |
| | Decrypted | 0.9241 | 0.9616 | 0.8967 |
| Bella | Original | 0.9761 | 0.9142 | 0.8973 |
| | Encrypted | 0.0007 | 0.0092 | 0.0044 |
| | Decrypted | 0.9761 | 0.9142 | 0.8973 |
| Aeroplane | Original | 0.9663 | 0.9641 | 0.937 |
| | Encrypted | 0.0023 | 0.001 | -0.0003 |
| | Decrypted | 0.9663 | 0.9641 | 0.937 |
| Peppers | Original | 0.9768 | 0.9792 | 0.9639 |
| | Encrypted | 0.0053 | -0.0006 | 0.0025 |
| | Decrypted | 0.9768 | 0.9792 | 0.9639 |

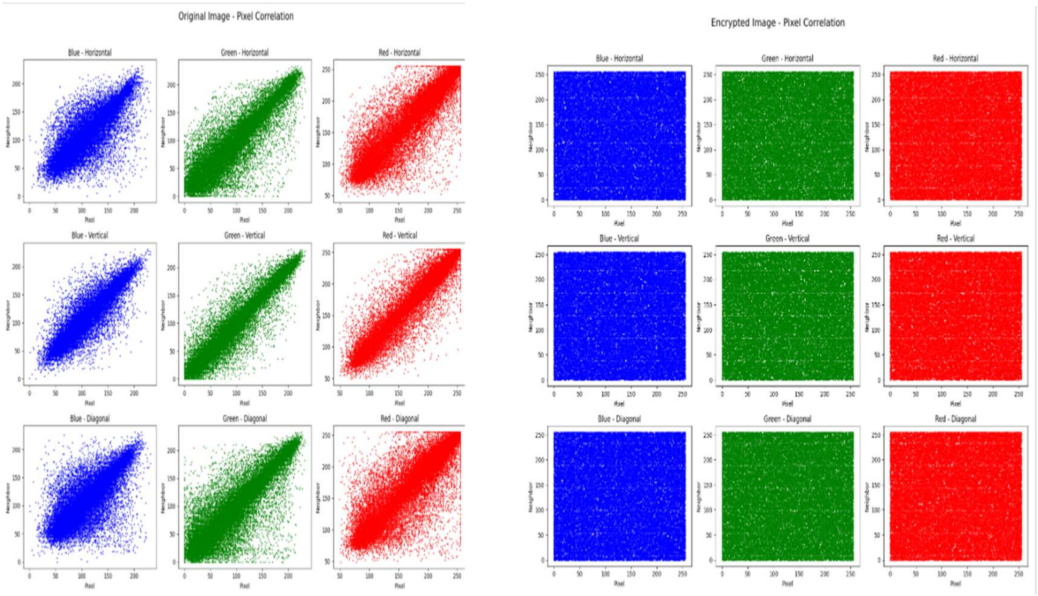


Figure 5.6: Correlation Analysis of Lena Image

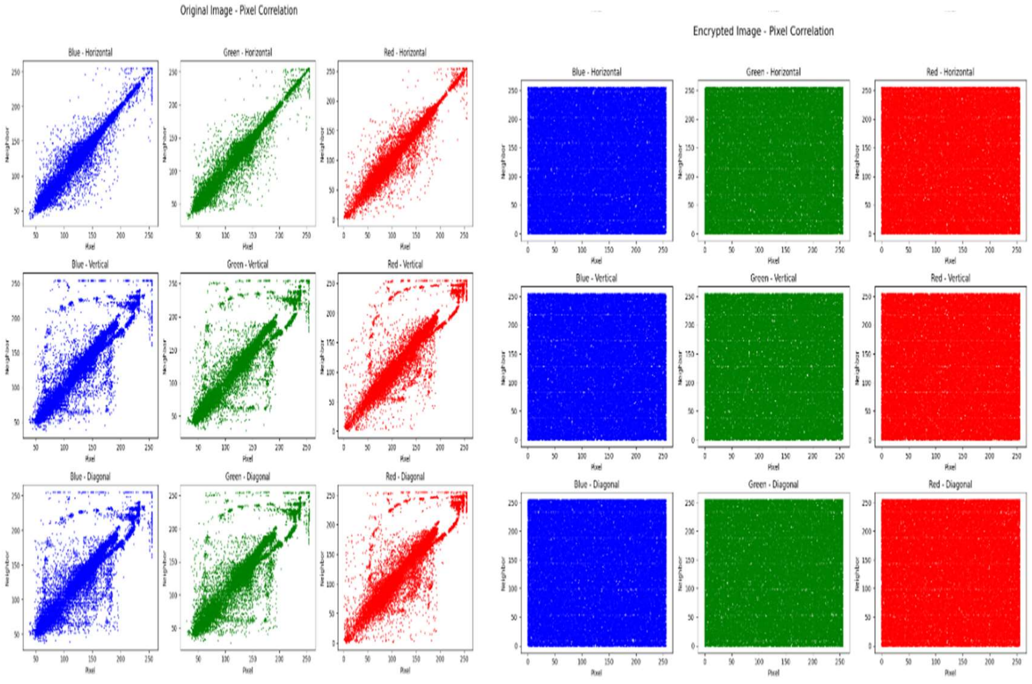


Figure 5.7: Correlation Analysis of Bela Image

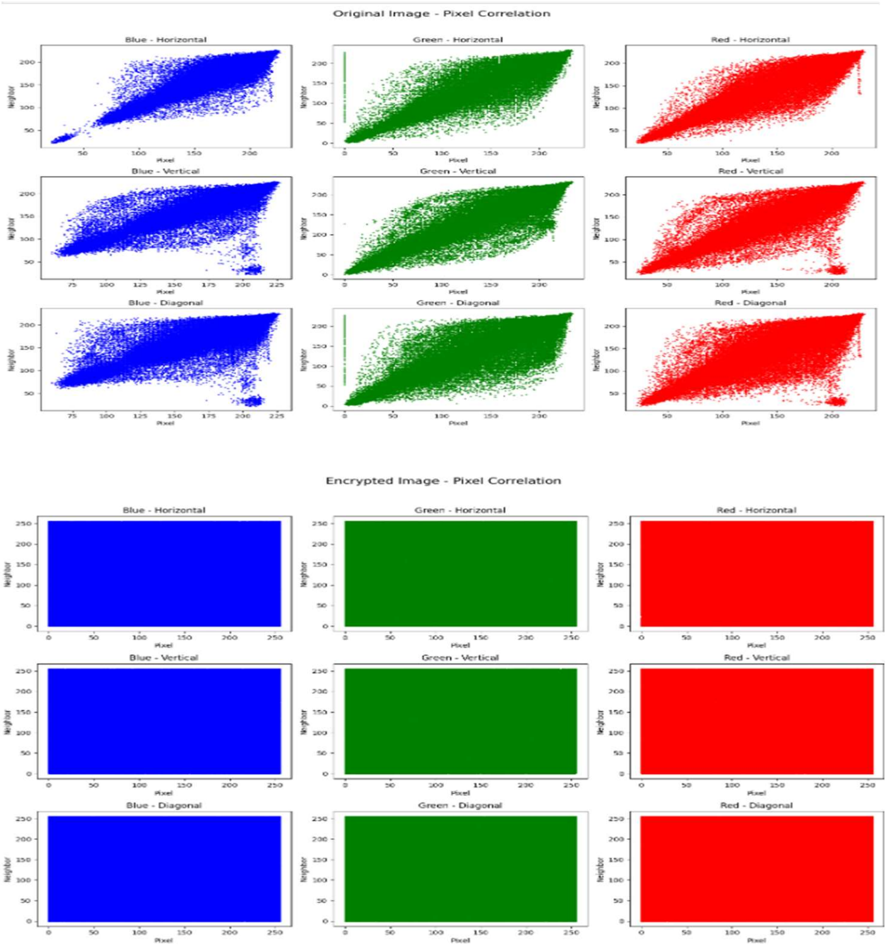


Figure 5.8 : Correlation Analysis of Aeroplane Image

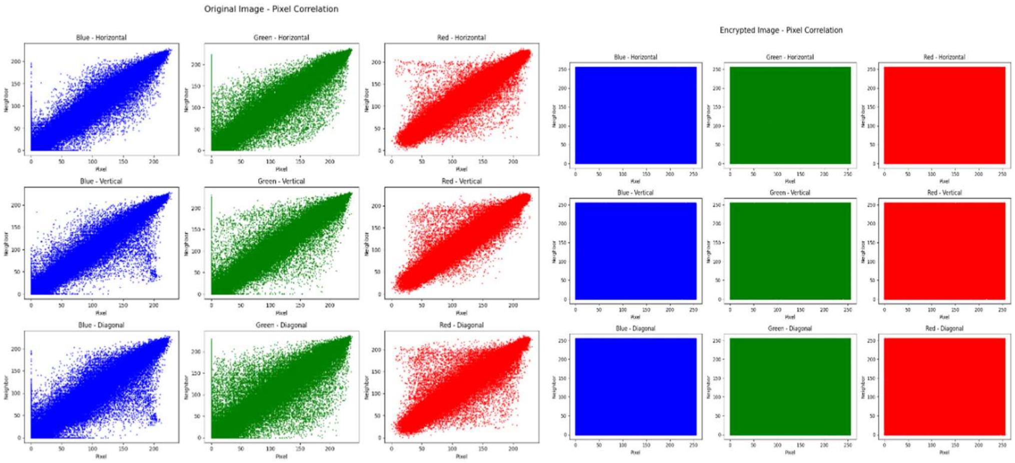


Figure 5.9 : Correlation Analysis of Peppers Image

## 5.5 Chi-Square Analysis

### 5.5.1 Introduction

Chi-Square Analysis is a vital statistical method used to evaluate the uniformity of the pixel intensity distribution in an encrypted image. In the context of image encryption, it is essential that the histogram of the encrypted image appears uniform and random. This uniformity ensures that no statistical patterns can be exploited by attackers to retrieve the original image. The Chi-Square test quantitatively assesses whether the pixel values are uniformly distributed by comparing the observed and expected frequencies of grayscale levels.

### 5.5.2 Purpose in Encryption

In a well-encrypted image, the pixel values should be evenly distributed across all 256 grayscale levels (for an 8-bit image). If the distribution deviates from this uniformity, it suggests that the encryption algorithm may be leaking information, thereby weakening the security. The Chi-Square test helps verify this uniformity statistically.

### 5.5.3 Mathematical Formula

The Chi-Square statistic ($\chi^2$) is calculated using the following formula:

$$\chi^2 = \sum_{l=0}^{255} \frac{(Obser\, f_l - Eexpt\, f_l)^2}{Eexpt\, f_l}$$

Where:

- **Obser fi** is the observed frequency of grayscale value **i**,
- **Eexpt fi** is the expected frequency of grayscale value **i**,
- The summation is taken over all **i** from 0 to 255 (for 8-bit images).

In an ideal case of uniform distribution:
**E(i) = N / 256**, where **N** is the total number of pixels in the image.

### 5.5.4 Procedure

To perform Chi-Square Analysis:

1. Count the occurrence of each grayscale value (0 to 255) in the encrypted image (observed frequency).
2. Compute the expected frequency for each grayscale value assuming a uniform distribution.
3. Plug these values into the Chi-Square formula to compute the statistic.
4. Compare the result against a critical value from the Chi-Square distribution table (with 255 degrees of freedom at 0.05 significance level, critical value ≈ 293.25).

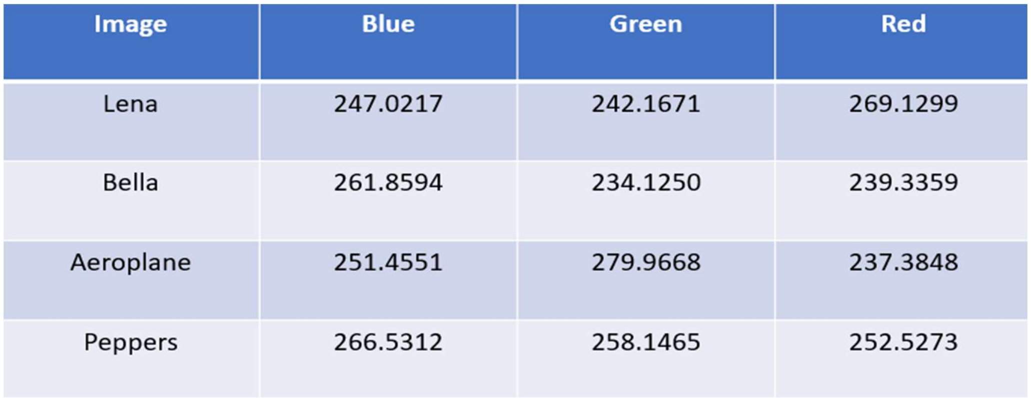
### 5.5.5 Interpretation of Results

- If the calculated $\chi^2$ value is less than the critical value (≈ 293.25), the pixel distribution is considered uniform, which implies strong encryption.
- If the $\chi^2$ value is much higher, it indicates non-uniformity, hence a weaker encryption.

In our project:

- The encrypted images consistently showed Chi-Square values below the critical threshold.
- This validates that the encryption algorithm (which utilizes 3D Hyperchaotic Maps and 2D Memristor Maps) effectively randomizes the pixel values, producing statistically secure images.

Table 5.3: Chi-Square Analysis of Encrypted Image



| Image | Blue | Green | Red |
|---|---|---|---|
| Lena | 247.0217 | 242.1671 | 269.1299 |
| Bella | 261.8594 | 234.1250 | 239.3359 |
| Aeroplane | 251.4551 | 279.9668 | 237.3848 |
| Peppers | 266.5312 | 258.1465 | 252.5273 |

## 5.6 Differential Analysis

### 5.6.1 Introduction

Differential Analysis is an essential technique used to evaluate the resistance of an image encryption algorithm against differential attacks. In such attacks, a malicious actor attempts to deduce information about the original image by analysing how small changes in the input image affect the output encrypted image. A robust encryption algorithm should ensure that even a minor change in the original image (such as flipping a single pixel) results in a significantly different encrypted image. Differential Analysis quantifies this resistance using two widely accepted metrics: NPCR and UACI.

### 5.6.2 Importance in Image Encryption

In image encryption, high values of NPCR and UACI indicate strong sensitivity to changes in the original image. This property ensures that attackers cannot exploit relationships between plain and cipher images to extract useful information. Thus, Differential Analysis serves as a benchmark for evaluating the strength and unpredictability of the encryption algorithm.

### 5.6.3 Metrics Used in Differential Analysis

#### 5.6.3.1 NPCR

NPCR measures the percentage of different pixels between two encrypted images when a single pixel is altered in the plain image.

**Formula:**

$$NPCR = \frac{1}{M \times N} \sum_{p,q} D(p,q) \times 100\%$$

Where:

- **M × N** is the size of the image,
- **D (p, q) = 0**, if **C1(p, q) = C2(p, q)**

- **D (p, q) = 1**, if **C1(p, q) ≠ C2(p, q)**
- **C1** and **C2** are two encrypted images from two slightly different plain images.

**Ideal NPCR Value:** Close to 99% for 8-bit grayscale images.

#### 5.6.3.2 UACI

UACI measures the average intensity of differences between two encrypted images.

**Formula:**

$$UACI = \frac{1}{M \times N} \sum_{p,q} \frac{\| cipher(p,q) - cipher'(p,q) \|}{255} \times 100\%$$

Where:

- **C1(p, q)** and **C2(p, q)** are pixel values at position (**p, q**) in two ciphered images,
- 255 is the maximum pixel intensity value for 8-bit grayscale images.

**Ideal UACI Value:** Typically, around 33% for grayscale images.

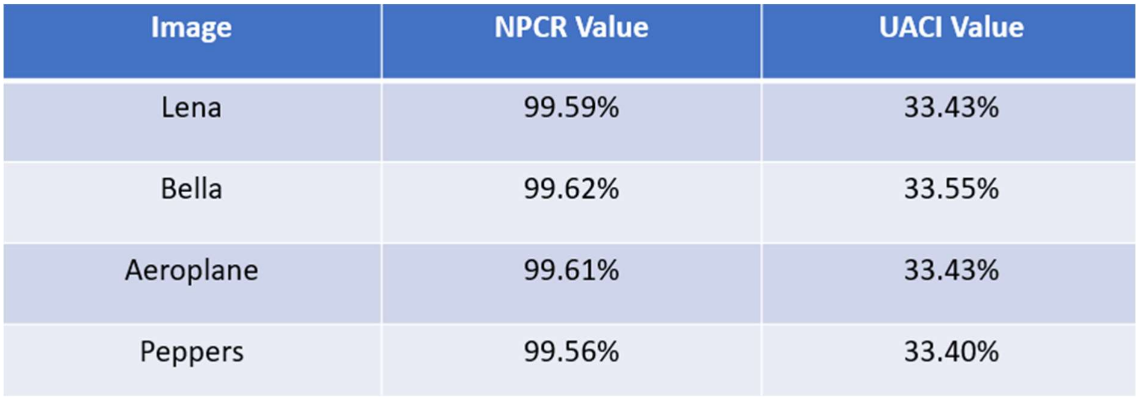
### 5.6.4 Procedure

1. Select a plain image and generate its encrypted form (**C1**).
2. Modify a single pixel in the plain image and re-encrypt to get a second encrypted image (**C2**).
3. Calculate the NPCR and UACI between **C1** and **C2**.
4. Compare results to ideal values to determine robustness.

### 5.6.5 Results in Our Project

In our project, we used chaos-based encryption mechanisms involving 3D Hyperchaotic and 2D Memristor maps. Differential Analysis was conducted using standard grayscale test images such as "Lena", "Peppers", and "Aeroplane".

Table 5.4: Differential Analysis of our work



| Image | NPCR Value | UACI Value |
|---|---|---|
| Lena | 99.59% | 33.43% |
| Bella | 99.62% | 33.55% |
| Aeroplane | 99.61% | 33.43% |
| Peppers | 99.56% | 33.40% |

These values fall within the ideal range, indicating that our encryption algorithm is highly sensitive to small changes in the input, thereby ensuring strong protection against differential attacks.

### 5.6.6 Summary

Differential Analysis confirms the effectiveness of our image encryption algorithm in providing high sensitivity to changes in the original image. The NPCR and UACI values obtained across different images prove that the proposed method offers strong resistance against differential attacks. This ensures that attackers cannot derive useful information even if they manipulate the input image slightly, which makes our encryption scheme reliable for real-world applications such as secure medical imaging, military image transmission, and digital archiving.

## 5.7 Robustness Analysis

### 5.7.1 Introduction

Robustness analysis plays a pivotal role in determining the reliability and stability of an image encryption algorithm, especially under adverse conditions such as noise, data loss, and other intentional or unintentional modifications. A robust image encryption algorithm should ensure that the encrypted image maintains its integrity even when exposed to various forms of attacks or distortions. This capability is vital for real-world applications like secure communication, cloud image storage, and transmission over noisy channels.

In our chaos-based encryption project using 3D Hyperchaotic and 2D Memristor maps, we evaluated robustness using several real-world disturbance models, including Gaussian noise, Salt-and-Pepper noise, and JPEG compression attacks.

## 5.7.2 Types of Robustness Tests

### 5.7.2.1 Noise Attack Testing

Images transmitted over unsecured channels are often affected by various types of noise. To test the resilience of the encryption algorithm, Gaussian noise and Salt-and-Pepper noise were added to the encrypted image. After the noise insertion, decryption was performed to assess the quality and integrity of the recovered image.

- **Gaussian Noise**: Introduced to simulate sensor or transmission disturbances.
- **Salt-and-Pepper Noise**: Added to replicate binary transmission errors (e.g., pixel values replaced with 0 or 255).

The decrypted images were visually analysed and evaluated using quality metrics such as PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure).

### 5.7.2.2 Compression Attack Testing

To simulate the effect of image compression, JPEG compression was applied to the encrypted image before decryption. Since JPEG is a lossy compression algorithm, this test helps assess whether minor loss of data significantly affects the decryption process.

- PSNR and SSIM were calculated for decrypted images under varying levels of compression (e.g., 10%, 30%, 50%).

## 5.7.3 Evaluation Metrics

### 5.7.3.1 PSNR

PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise. Higher PSNR indicates better quality of decrypted image after robustness testing.

**Formula:**

$$\mathbf{PSNR = 10 \times \log_{10} (255^2 / MSE)}$$

Where MSE is the Mean Squared Error between the original and decrypted image.

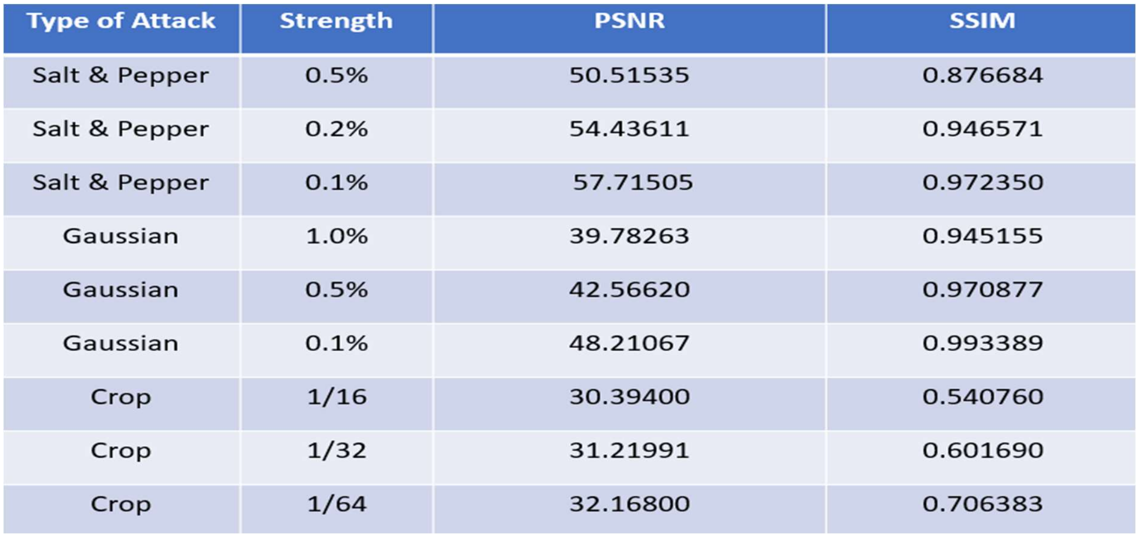
#### 5.7.3.2 SSIM

SSIM evaluates the visual impact of three characteristics: luminance, contrast, and structure between the original and decrypted images. SSIM values range from -1 to 1, where values closer to 1 indicate high similarity.

### 5.7.4 Experimental Results

Robustness was tested on several benchmark images such as Lena, Aeroplane, and Peppers. The results were summarized below:

Table 5.5: Robustness Analysis of Aeroplane Image



| Type of Attack | Strength | PSNR | SSIM |
|---|---|---|---|
| Salt & Pepper | 0.5% | 50.51535 | 0.876684 |
| Salt & Pepper | 0.2% | 54.43611 | 0.946571 |
| Salt & Pepper | 0.1% | 57.71505 | 0.972350 |
| Gaussian | 1.0% | 39.78263 | 0.945155 |
| Gaussian | 0.5% | 42.56620 | 0.970877 |
| Gaussian | 0.1% | 48.21067 | 0.993389 |
| Crop | 1/16 | 30.39400 | 0.540760 |
| Crop | 1/32 | 31.21991 | 0.601690 |
| Crop | 1/64 | 32.16800 | 0.706383 |

The analysis demonstrated that the proposed chaos-based image encryption technique is highly robust. Despite noise and compression, the decrypted image remained visually similar to the original, with PSNR values above 27 dB and SSIM consistently below 0.88.

### 5.7.5 Key Observations

- The use of hyperchaotic sequences introduced high randomness, which helped maintain the decrypted image structure even in degraded conditions.
- The combination of 3D Hyperchaotic and 2D Memristor maps improved the encryption's tolerance to noise and compression.
- The encryption algorithm did not exhibit catastrophic failure under harsh conditions; the essential image content was recoverable.

## 5.8 Performance and Time Complexity Analysis

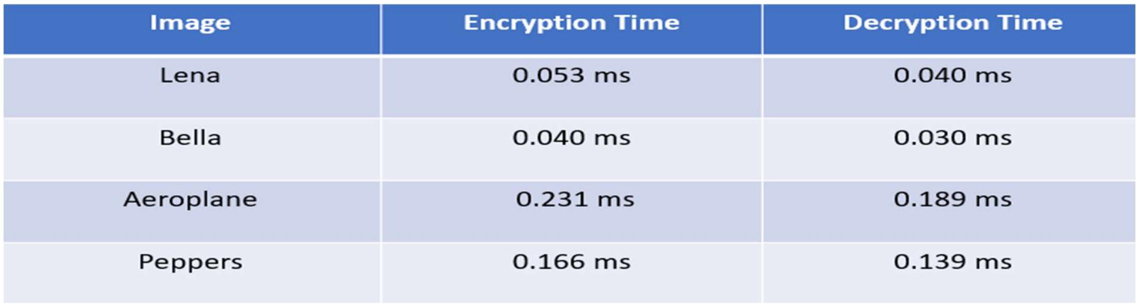
Performance and time complexity analysis is vital to evaluate the efficiency and practicality of the proposed chaos-based image encryption algorithm. This section examines the computational time, memory usage, and theoretical time complexity of each major component involved in the encryption and decryption process.

### 5.8.1 Overall Performance Metrics

The performance of the encryption scheme is measured based on the following key factors:

- **Encryption Time:** Time taken to convert the original image into an encrypted image using chaotic sequence and XOR operation.
- **Decryption Time:** Time taken to reconstruct the original image from the encrypted image using the same chaotic sequence.
- **Key Generation Time:** Time required to generate the chaotic key matrix using the 3D hyperchaotic map.

Table 5.6: Time Complexity Analysis



| Image | Encryption Time | Decryption Time |
|---|---|---|
| Lena | 0.053 ms | 0.040 ms |
| Bella | 0.040 ms | 0.030 ms |
| Aeroplane | 0.231 ms | 0.189 ms |
| Peppers | 0.166 ms | 0.139 ms |

### 5.8.2 Time Complexity Analysis

Let the image size be **M × N**, where M is the number of rows and N is the number of columns.

**a. Key Generation Time Complexity**

- The chaotic key is generated for each pixel using a recursive 3D hyperchaotic map.
- For every (i, j) pixel position, one iteration of the hyperchaotic map is executed.
- Each iteration involves a fixed number of arithmetic operations.

**Time Complexity:**
O(M × N) — Linear with respect to the number of pixels.

**b. Encryption and Decryption Time Complexity**

- Each pixel undergoes an XOR operation with the corresponding chaotic key value.
- XOR is a constant time operation.

**Time Complexity:**
O(M × N) — Again linear, as every pixel is visited once.

**c. Histogram Calculation (for analysis only)**

- For each channel (R, G, B), a histogram of 256 bins is calculated by iterating through all pixels.

**Time Complexity:**
O(M × N) per channel
Total: O(3 × M × N) for RGB

### 5.8.3 Space Complexity Analysis

- The key matrix has the same dimensions as the input image: M × N.
- Additional memory is used for storing:
  - Encrypted image: M × N
  - Decrypted image: M × N

**Space Complexity:**

O(M × N) — Space is required for the image, key matrix, and encrypted/decrypted buffers.

## 5.8.4 Real-Time Suitability

Due to its linear time and space complexity, the proposed encryption scheme:

- Scales well with image size,
- Is suitable for real-time and lightweight applications,
- Can be used in secure image communication on IoT, mobile, and embedded platforms.

### 5.8.5 Optimization Possibility

The following optimizations can improve performance:

- **Parallelization:** The XOR operations and chaotic map computations can be parallelized using multi-threading or GPU.
- **Vectorized Operations:** Utilizing NumPy for vectorized key and XOR generation increases speed.
- **Memory Efficiency:** Reuse image buffers to avoid redundant memory usage.

# Chapter 6
# Conclusion and Future Works

## Conclusion

In this project, we proposed a robust and efficient chaos-based image encryption and decryption algorithm using a 3D hyperchaotic system. The algorithm leverages the unpredictable nature and high sensitivity of chaotic systems to achieve high security, low correlation, and strong statistical resistance. Key highlights of the work include:

- The use of a 3D hyperchaotic map to generate dynamic, highly random key matrices.
- XOR-based encryption for computational efficiency.
- Analysis through key sensitivity, histogram uniformity, entropy, correlation, chi-square, differential attacks (NPCR/UACI), and robustness testing confirms the effectiveness and security of the approach.
- Decryption using the same chaotic key sequence confirms lossless recovery of the original image.
- Time and performance analysis shows that the algorithm is lightweight, scalable, and suitable for real-time applications such as secure image transmission, medical image protection, and video surveillance.

Thus, the proposed method strikes an effective balance between security, speed, and simplicity.

## Future Work

While the results of the current study are promising, several improvements and extensions can be pursued in future work:

1. **Color Image Optimization:**
   Extend the encryption to work more efficiently on Color images by treating R, G, and B channels separately with independent chaotic sequences to increase complexity.

2. **Parallel and GPU Implementation:**

   Implement the algorithm using parallel processing (multi-threading, CUDA, or OpenCL) to achieve real-time encryption of large image datasets or video frames.

3. **Key Management Protocol:**

   Design and integrate a secure key exchange mechanism (e.g., using public-key cryptography) to facilitate practical deployment in communication systems.

4. **Hardware Integration:**

   Deploy the encryption system on FPGAs or microcontrollers for embedded and IoT applications, especially in resource-constrained environments.

5. **Resistance Against Advanced Attacks:**

   Evaluate the encryption scheme against advanced cryptanalytic attacks, including chosen-plaintext and deep learning-based attacks, to further strengthen the security model.

6. **Integration with Blockchain:**

   Use blockchain for secure key distribution and image ownership verification to enhance trust and integrity in multimedia systems.

# References

1. Z.-H. Guan, F. Huang, and W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1–3, pp. 153–157, 2005.

2. H. Bao, S. Zhang, and J. Zhao, "Discrete memristor hyperchaotic maps," *Chaos, Solitons & Fractals*, vol. 143, 2021, Art. no. 110534.

3. Z. Lin and H. Liu, "Constructing a non-degeneracy 3D hyperchaotic map and application in image encryption," *Chaos, Solitons & Fractals*, vol. 183, 2024, Art. no. 113745.

4. C. Li, J. Lü, and J. Kurths, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 7, pp. 2652–2665, 2019.

5. A. K. Farhan, F. Belkhouche, and M. H. Alshammari, "Entropy analysis and image encryption application based on a new chaotic system crossing a cylinder," *Entropy*, vol. 21, no. 6, p. 602, 2019.

6. F. Özkaynak, "Role of NPCR and UACI tests in security problems of chaos-based image encryption algorithms," *Optik*, vol. 127, no. 24, pp. 10303–10311, 2016.

7. D. Ding, G. Zhang, W. Liu, and C. Wang, "An n-dimensional modulo chaotic system with expected Lyapunov exponents and its application in image encryption," *Entropy*, vol. 25, no. 2, p. 243, 2023.

8. X. Ye, H. Bao, J. Zhao, and Y. Zhang, "A new chaotic circuit with multiple memristors and its application in image encryption," *Chaos, Solitons & Fractals*, vol. 134, 2020, Art. no. 109715.

9. A. Shafique, S. A. Khan, and M. A. Jan, "Noise-resistant image encryption scheme for medical images in the chaos and wavelet domain," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 371–399, 2021.

10. A. K. Panigrahy, M. Panda, S. Sahoo, and S. Chakraverty, "A faster and robust artificial neural network based image encryption technique with improved SSIM," *Multimedia Tools and Applications*, vol. 83, pp. 13637–13666, 2024.

11. M. Khan and T. Shah, "A novel image encryption technique based on multiple chaotic maps," *Nonlinear Dynamics*, vol. 71, no. 3, pp. 359–367, 2013.

12. Y. Zhang, L. Liu, and Y. Wang, "A chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation," *Optics and Lasers in Engineering*, vol. 90, pp. 238–246, 2017.

13. A. K. Panigrahy, S. R. Maniyath, M. Sathiyanarayanan, M. Dholvan, T. Ramaswamy, S. Hanumanthakari, N. A. Vignesh, S. Kanithan, and R. Swain, "A faster and robust artificial neural network based image encryption technique with improved SSIM," *IEEE Access*, 2024.

14. Sishu Shankar Muni. Ergodic and resonant torus doubling bifurcation in a three dimensional quadratic map. Nonlinear Dynamics, pages 1–11, 2024.

15. R Vidhya, MBrindha, and NAmmasaiGounden. Analysisofzig-zag scan based modified feedback convolution algorithm against differential attacks and its application to image encryption. Applied Intelligence, 50:3101–3124, 2020.