International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 583-609

https://doi.org/10.70917/ijcisim-2025-0037
Article

An Image Encryption Scheme based on the 2D Hybrid Hyperchaotic Modified Lemniscate Map

Jackson J. and Perumal R. *

Department of Mathematics, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur 603203, Tamilnadu, India; jj3375@srmist.edu.in
* Correspondence author: perumalr@srmist.edu.in

Received date: 2 September 2024; Accepted date: 25 August 2025; Published online: 31 December 2025

Abstract: Secure image transmission is critical in today's digital landscape. Chaotic systems, renowned for their sensitivity to initial conditions and randomness, are widely applied in encryption schemes but often encounter limitations such as weak chaos and predictable structures. To address these challenges, this paper proposes a novel image encryption scheme based on a 2D Hybrid Hyperchaotic Modified Lemniscate (2D-HHML) map. The enhanced 2D-HHML map demonstrates superior chaotic properties, evidenced by a higher maximum Lyapunov exponent of 17.350 and increased sample entropy compared to conventional chaotic maps, indicating stronger randomness and unpredictability. The encryption framework integrates a selective pixel-shuffling strategy, which disrupts the spatial relationships between adjacent pixels, and a pixel-wise diffusion process, which further enhances security by masking statistical patterns in the encrypted image. The security and performance of the proposed scheme are rigorously evaluated through multiple statistical metrics, including NPCR, UACI, Shannon entropy, correlation coefficients, encryption time, and key sensitivity. Simulation results show that the scheme achieves an average NPCR of 99.6% and an average UACI of 33.39%, with Shannon entropy values approaching the ideal value of 8, low correlation coefficients close to zero, and efficient encryption times. Additionally, resistance against various differential and statistical attacks is validated through simulated attack analyses. Overall, the results confirm that the proposed encryption method provides high security, strong key sensitivity, and excellent performance, making it a promising candidate for secure transmission of sensitive image data.

Keywords: hyper chaotic map; lemniscate-based approach; pixel shuffle; diffusion mechanism

1. Introduction

The rapid expansion of online applications and communication technologies has led to the transmission of massive amounts of multimedia data over the Internet, making the security of digital images a critical concern. Cryptography provides a solution by converting sensitive information into an unintelligible format, ensuring that only authorized recipients can recover the original data [1]. Unlike textual data, digital images exhibit unique characteristics such as high redundancy, large data volumes, and strong correlations between neighboring pixels [2,3] that necessitate specialized encryption techniques [4,5]. This study introduces a novel chaotic system: the two-dimensional Hybrid Hyperchaotic Modified Lemniscate (2D-HHML) map. Building upon the well-known Lemniscate map, the 2D-HHML map is designed to exhibit stronger chaotic behavior, with significantly higher Lyapunov exponents and improved randomness properties. Its hyperchaotic nature ensures greater sensitivity to initial conditions and mitigates the risks associated with numerical degradation. Compared to existing chaotic maps, the 2D-HHML map offers enhanced unpredictability, a broader chaotic range, and superior statistical performance, as verified through bifurcation analysis, time series evaluation, sample entropy calculations, and the NIST SP 800-22 randomness tests. In the proposed image encryption scheme, the

2D-HHML map is employed to generate key matrices for confusion and diffusion processes. The encrypted images are rigorously evaluated for security and performance, demonstrating strong resistance to statistical and differential attacks and achieving high efficiency. The key contributions of this work are summarized as follows:

- **Development of a New Chaotic Map:** A 2D-HHML map with higher Lyapunov exponents and improved entropy, ensuring stronger chaotic behavior suitable for encryption applications.
- **Enhanced Image Encryption Scheme:** Integration of selective pixel-shuffling and pixel-wise diffusion strategies to maximize confusion and diffusion properties.
- **Comprehensive Security Analysis:** Extensive evaluations including NPCR, UACI, Shannon entropy, correlation coefficients, encryption time, and simulated attack resistance, confirming the robustness of the proposed method.
- **Improved Resistance to Finite-Precision Effects:** Ensuring stable chaotic behavior even under digital computational constraints, overcoming limitations faced by many traditional maps.

The remainder of this paper is organized as follows. Section 2 presents the literature review. Section 3 presents the methodology used wherein the proposed 2D-HHML map is studied. Section 4 details the image encryption algorithm based on the proposed map. Section 5 discusses the security and performance analysis. Section 6 concludes the paper with key findings and future research directions.

2. Literature Review

In recent decades, chaotic systems have gained substantial attention in the field of image cryptography due to their properties of non-periodicity, pseudo-randomness high sensitivity to initial conditions, and ergodicity [6]. Since Matthews' pioneering work in 1989 [7], numerous encryption methods have been developed based on chaotic dynamics [8–11], including the extension of one-dimensional chaotic maps to higher dimensions [12,13] and the combination of multiple maps to create more complex behaviors [14,15]. For example, Tutueva et al. [16] proposed adaptive symmetry in chaotic maps to enhance the chaotic range, while Fridrich [17] introduced a cipher based on the 2D Baker map. Similarly, Zarebnia et al. [18] presented a multiple-image encryption method using hybrid chaotic systems derived from the Arnold cat map. However, despite these advances, many chaotic-based encryption schemes exhibit significant limitations that compromise security. The chaotic generators used often produce relatively weak chaos, characterized by low positive Lyapunov exponents [19], resulting in reduced sensitivity to initial conditions and predictable behaviors. In 2024, Jackson and Perumal [20] proposed image encryption schemes based on a fractional ordered chaotic map [21] and 2D-Hyperchaotic Sine Logistic map [22]. Also, Ponmaheshkumar and Perumal [23] proposed an image encryption scheme based on 1D-Cosine Arcsine chaotic map. Furthermore, chaotic behavior suppression and degradation especially in simple one-dimensional maps like the logistic and tent maps due to finite-precision effects in digital computation [24], further weaken the randomness essential for strong encryption. Such weaknesses increase vulnerability to statistical attacks, reduce diffusion and confusion strength, and make encrypted images more susceptible to partial recovery by attackers. Although some countermeasures, such as improved precision methods or switching between multiple chaotic outputs, have been proposed [25], they add complexity without fully resolving the inherent limitations. In this section we recall the 2D-Traditional Lemniscate map (2D-TLM) as presented in [26]. The basic Lemniscate function is defined as,

$$\begin{cases} x(n+1) = \dfrac{\cos\left(2^r y(n)\right)}{1 + sin^2\left(2^r y(n)\right)} \\ y(n+1) = \dfrac{2^{1.5}\sin\left(2^r x(n)\right)\cos\left(2^r x(n)\right)}{1 + sin^2\left(2^r x(n)\right)} \end{cases} \tag{1}$$

where, x0 and y0 are the initial conditions and r is the control parameter.

The recurrence relations involve trigonometric functions cos and sin with arguments scaled by a factor of 2^r. These trigonometric functions map the inputs to periodic outputs, resulting in complex and potentially chaotic behavior over iterations. This map represents a discrete dynamical system. By iterating the map, one can study the evolution of points (x_n, y_n) in the plane. Dynamical systems like this can exhibit various behaviors, including fixed points, periodic orbits, and chaotic trajectories, depending on the parameter r and initial conditions. The denominators $1 + sin^2(2^r y_n)$ and $1 + sin^2(2^r x_n)$ serve as normalization factors ensuring that the outputs remain bounded. These factors prevent the outputs from growing unbounded, a common feature in maps aiming to produce bounded, complex dynamics. The parameter r controls the scaling of the arguments of the trigonometric functions. Different values of r can significantly alter the behavior of the map, potentially leading to bifurcations

and changes in the system's stability and periodicity. In [26] Saber et al. presented a modified version of the Lemniscate map called 2D-Practical Lemniscate Chaotic (2D-PLC) map as follows,

$$\begin{cases} x(n+1) = \dfrac{\cos\left(2^r y(n)\right)}{2 - \cos^2\left(2^r y(n)\right)} \\ y(n+1) = \dfrac{2^{0.5}\sin\left(2 \times 2^r x(n)\right)}{1 + \sin^2\left(2^r x(n)\right)} \end{cases} \tag{2}$$

In 2021, Teng et al. [27] proposed the following Sine Logistic Modulation (2D-SLM) map,

$$\begin{cases} x(i+1) = \sin\left(l/\sin\left(y(i)\right)\right) \\ y(i+1) = m\sin\left(\pi\left(x(i) + y(i)\right)\right) \end{cases} \tag{3}$$

where l and m are the control parameters.

3. Methodology

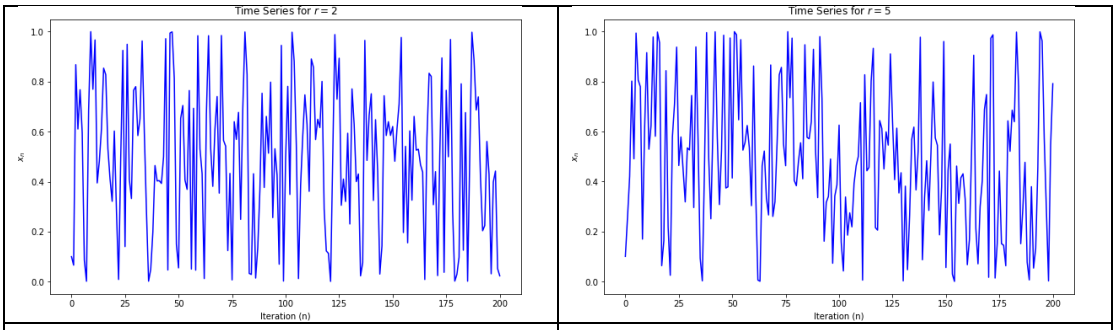
In this section, we propose the 2D-Hybrid Hyperchaotic Modified Lemniscate (2D-HHML) map as given in Equation 4,

$$\begin{cases} x(n+1) = \dfrac{\sin\left(\dfrac{3^r \cos\left(2^r x(n)\right)}{\sin^2\left(2^r y(n)\right)}\right)}{1 + \cos^2\left(\dfrac{3^r \cos\left(2^r x(n)\right)}{\sin^2\left(2^r y(n)\right)}\right)} \\ y(n+1) = \dfrac{\cos\left(\dfrac{3^r \cos\left(2^r y(n)\right)}{\sin^2\left(2^r x(n)\right)}\right)}{1 + \sin^2\left(\dfrac{3^r \cos\left(2^r y(n)\right)}{\sin^2\left(2^r x(n)\right)}\right)} \end{cases} \tag{4}$$

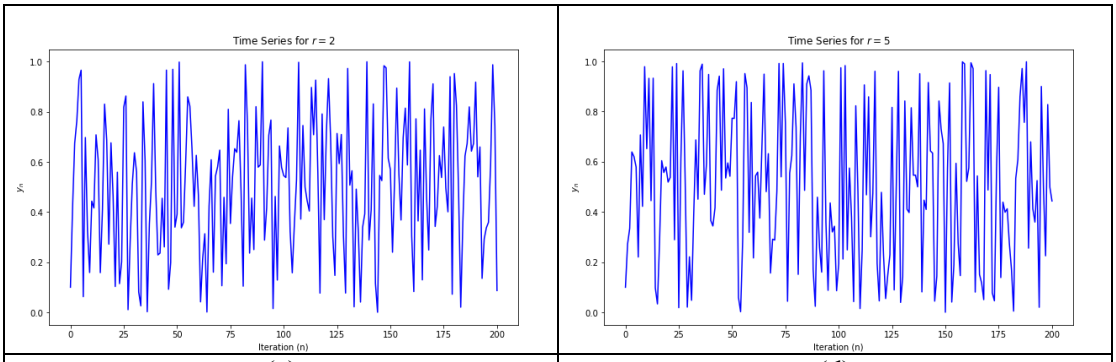
The values x_0 and y_0 are the initial conditions and r is the control parameter of this 2D-HHML map. This newly proposed map is used to generate the secret key which is used to encrypt and decrypt in the image encryption process. We investigate the significant nonlinear dynamics exhibited by this chaotic map through analyses including time series plots, bifurcation diagrams, and the maximum Lyapunov exponent plot. We illustrate that this system features a prominently large positive maximum Lyapunov exponent and a broad parameter range r where the map displays intricate chaotic behavior.

3.1. Time Series Analysis

The initial condition $x_0 = 0.1$ and $y_0 = 0.1$ are fixed constant, and Figure 1 illustrates time series plots of the output from the proposed map for various values r. It is observed that the new map exhibits a series of period-doublings within a narrow range of r. As a result, chaotic dynamics are observed over a broad range of r values.



(a) (b)

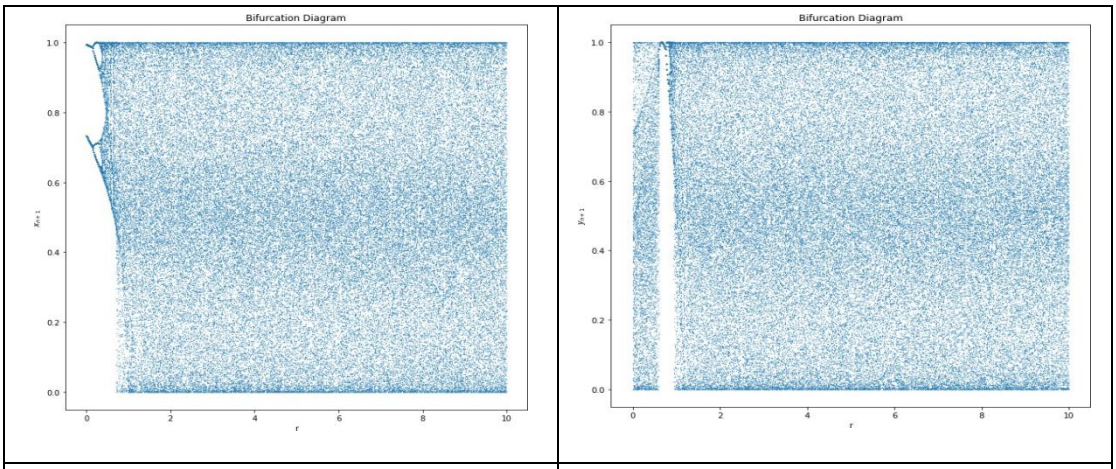


(c) (d)

Figure 1. Time series of (a) x_n for $r = 2$ (b) x_n for $r = 5$ (c) y_n for $r = 2$and (d) y_n for $r = 5$.

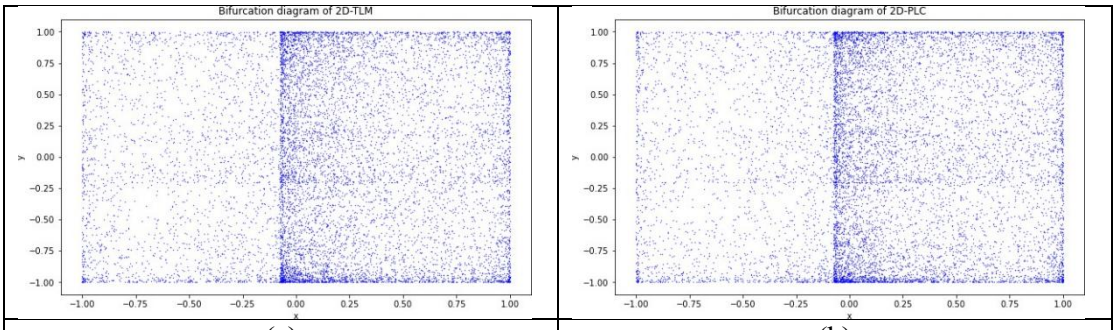
3.2. Bifurcation Diagram

The bifurcation diagram is created by varying the parameter r over a range of values. For each value of r, the long-term behavior of the system is observed. As r is varied, the system may exhibit bifurcations (splitting of fixed points), leading to periodic doubling and eventually to chaotic behavior. In chaotic regimes, the diagram shows a dense, intricate pattern of points indicating the complex, non-repeating behavior of the system. The bifurcation of our chaotic map is given in Figure 2. The comparison of bifurcation with 2D-TLM and 2D-PLC is given in Figure 3.

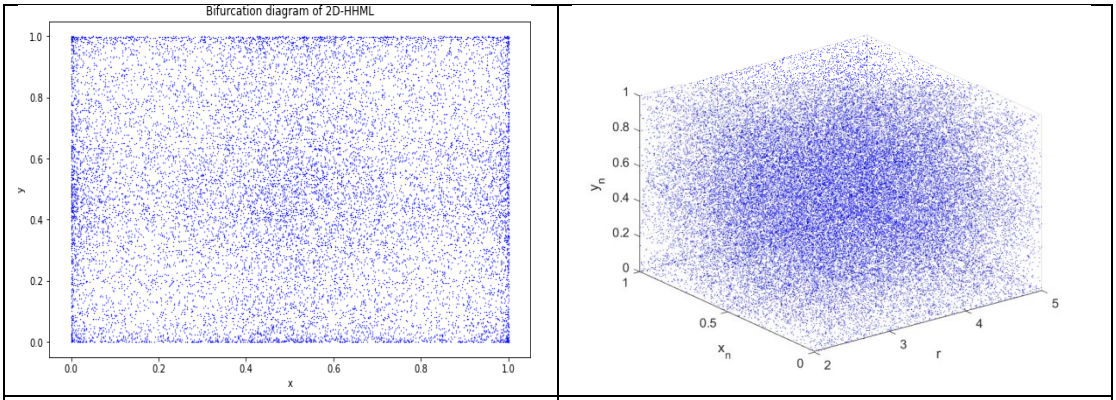


(a) (b)

Figure 2. Bifurcation of (a) x_n with $r \in (0, 10)$ (b) y_n with $r \in (0, 10)$.



(a) (b)



(c) (d)

Figure 3. Comparison of Bifurcation on (a) 2D-TLM, (b) 2D-PLC (c) 2D-HHML with $r = 3$ and (d) 3D bifurcation diagram of 2D-HHML with $r \in (2, 5)$.

3.3. Lyapunov exponent

The Lyapunov exponent (LE) is used to measure the rate at which nearby trajectories in a system diverge. It can be used to analyze and ensure the security and effectiveness of an encryption algorithm which is based on chaotic systems. Consider a 2D discrete-time dynamical system given by the map:

$$x_{n+1} = f(x_n) \tag{5}$$

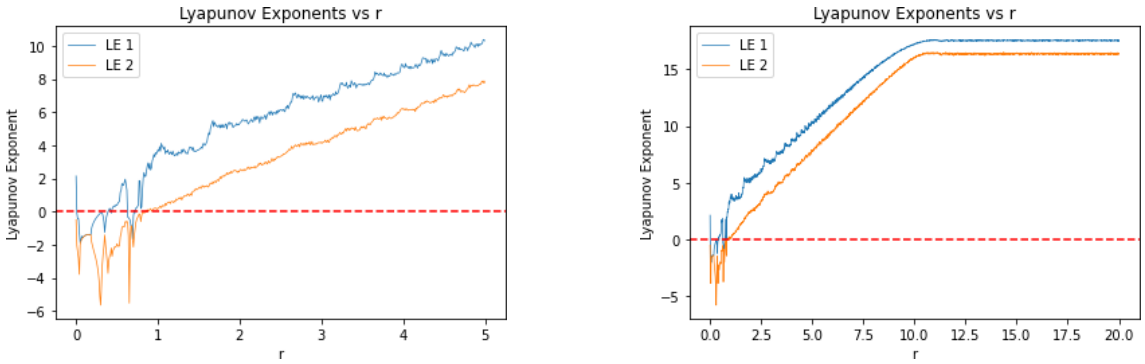
where $\boldsymbol{X}_n = (x_n, y_n)^T$ represents the state of the system at time step n and f is a nonlinear function. To compute the Lyapunov exponent, we need to evaluate the Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_n)$ of the map f at each point $\boldsymbol{x}_n$.

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} \tag{6}$$

The Lyapunov exponents λ1 and λ2 are defined as:

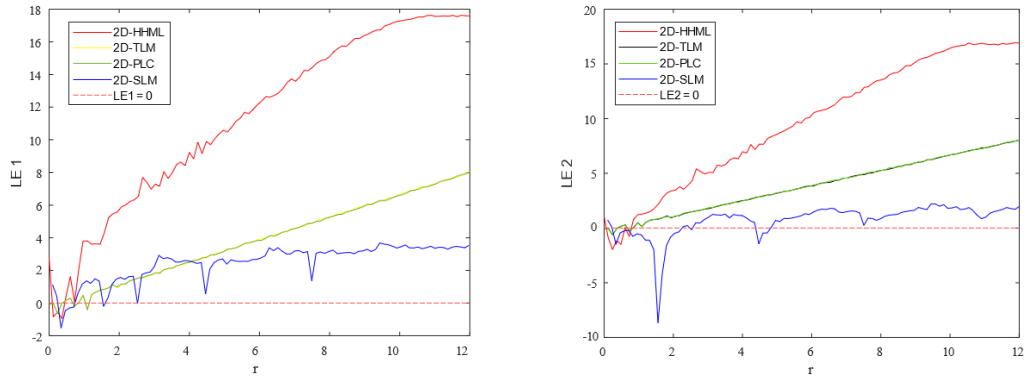
$$\lambda_i = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \ln\left|\mu_i\left(J(x_k)\right)\right| \tag{7}$$

where μ_i are the eigenvalues of the Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_n)$ at the k-th iteration. For a chaotic system, at least one of the Lyapunov exponents is positive, indicating sensitivity to initial conditions. Our 2D-HHML map shows the highest Lyapunov exponent of 17.350 at $r = 10.235$. Figure 4 displays the Lyapunov exponent spectra of the proposed chaotic map and indicates that the Lyapunov exponents have significantly large positive values, confirming the presence of complex behavior in the proposed map. Also, the Lyapunov exponent is compared with some well-known maps in Figure 5. A positive Lyapunov exponent indicates chaotic behavior, meaning that small differences in initial conditions lead to exponentially diverging outcomes. This property is crucial for encryption as it ensures that even a tiny change in the key will result in a completely different encrypted image. A higher Lyapunov exponent signifies a greater level of chaos, which generally results in a more secure encryption method. This increased chaos makes the encrypted image more sensitive to key variations and more challenging to predict or reverse-engineer.



(a) (b)

Figure 4. The Lyapunov exponents LE 1 and LE 2 for (a) $r \in (0, 5)$ and (b) $r \in (0, 20)$.

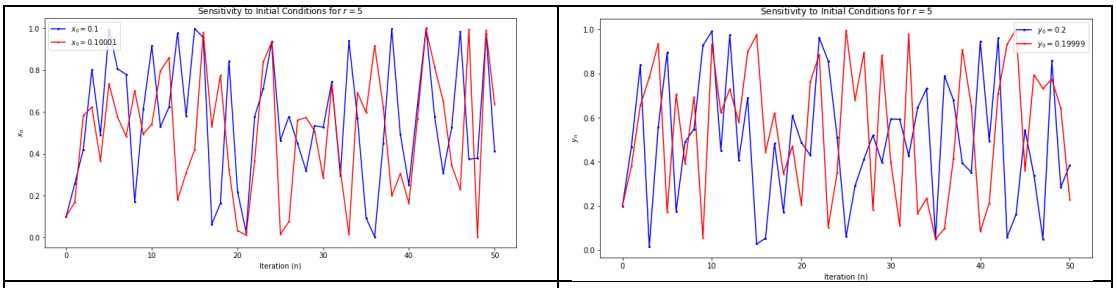


(a) (b)

Figure 5. Comparison of (a) LE 1 and (b) LE 2 for $r \in (0, 12)$.

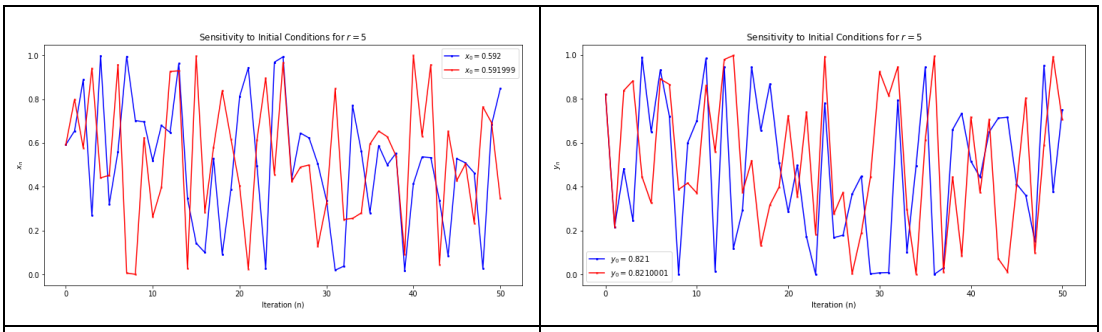
3.4. Sensitivity to initial condition

One of the defining characteristics of chaotic systems is their sensitivity to initial conditions. This means that chaotic systems are extremely responsive to small changes in their starting state, where even minor variations can lead to completely different chaotic outcomes. The sensitivity is quantified by measuring the correlation between two data sequences. For instance the control parameter r is fixed at 5 and the sensitivity is displayed for two cases. First the time series is plotted for the values $(x_0, y_0) = (0.1,0.2)$ and $(x_0, y_0) = (0.10001,0.19999)$. From Figure 6 it can be seen that the time series varies rapidly as the iteration increases. Similarly, the time series is plotted for the values $(x_0, y_0) = (0.592,0.5919999)$ and $(x_0, y_0) = (0.821,0.8210001)$ with $r = 5$ in Figure 7. It can be observed that the 2D-HHML chaotic map is highly sensitivity to initial condition. The figures reveal that as the number of iterations increases, even minor changes in initial values or control parameters can result in entirely different chaotic trajectories.



(a) Time series of x_n (b) Time series of y_n

Figure 6. Sensitivity to initial conditions (0.1,0.2) and (0.10001,0.19999).



(a) Time series of x_n (b) Time series of y_n

Figure 7. Sensitivity to initial conditions (0.592,0.821) and (0.5919999,0.8210001).

4. The proposed encryption scheme

This section presents an image encryption scheme utilizing the 2D-HHML chaotic map, which consists of three primary components: generation of secret keys and chaotic sequences, and confusion and diffusion processes accompanied by flowcharts for a step-by-step illustration of each procedure in Figure 8.

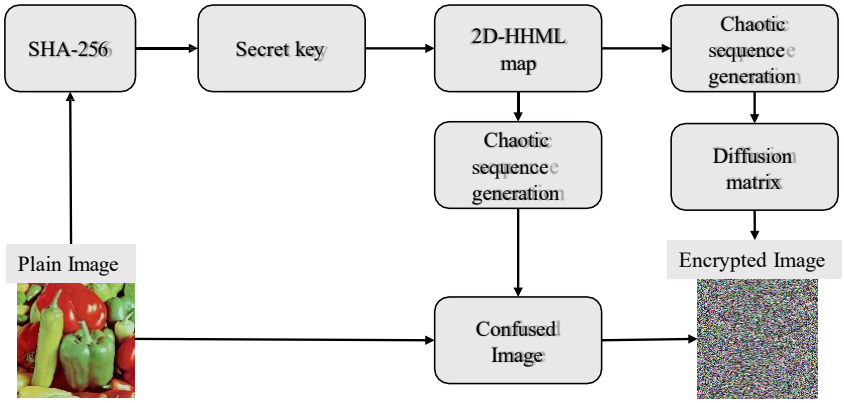


Figure 8. Flowchart of the encryption scheme.

4.1. Secret Key Generation using SHA-256

In the proposed image encryption scheme, the secret key is derived from the input image using the SHA-256 cryptographic hash function. This ensures high sensitivity, uniqueness, and resistance to brute-force and differential attacks. The secret key is then used to determine the initial conditions and the single control parameter of the 2D Hybrid Hyperchaotic Modified Lemniscate Map. The detailed key generation steps are as follows:

- **Image Preprocessing:** The input image I is first converted into a one-dimensional array by concatenating the pixel values of all three RGB channels.
- **SHA-256 Hashing:** The SHA-256 algorithm is applied to the pixel sequence to produce a 256-bit (64-character hexadecimal) hash value:

$$H = SHA - 256(I)$$

- **Hash Segmentation:** The hash string H is divided into three segments, each containing 16 hexadecimal characters:

$$H = H1 \parallel H2 \parallel H3$$

- **Hexadecimal to Decimal Conversion:** Each segment is converted from hexadecimal to a decimal number:

$$D_i = hex2dec(Hi), i = 1,2,3$$

- **Normalization:** Each decimal number D_i is normalized to the interval (0, 1) to obtain the initial conditions x_0, y_0, and control parameter r:

$$x_0 = \frac{D_1 mod 10^8}{10^8}, y_0 = \frac{D_2 mod 10^8}{10^8}, r = 4 + \frac{D_3 mod 10^8}{10^8}$$

- **Chaotic Initialization:** The values x_0, y_0, and r are used to initialize the 2D Hybrid Hyperchaotic Modified Lemniscate Map for generating chaotic sequences required in the encryption process.

This method ensures that the encryption process is tightly coupled with the image content, making the system highly sensitive to any changes in the input. Here the constant term 4 ensures that $r > 4$ making it fall within the most chaotic range. With these initial values and control parameter, we generate a chaotic sequence that matches the size of the plaintext image. This key plays a crucial role in generating the essential initial values and parameters necessary for the proper functioning of the proposed 2D-HHML map. Notably, this strategic integration not only enhances the algorithm's robustness but also contributes to the optimization of encryption speed. Each subsequent chaotic sequence begins with the final values of the preceding sequence. These chaotic matrices are reshaped and employed to scramble pixel positions (confusion) and modify pixel values (diffusion) in subsequent encryption stages.

4.2. Confusion Phase

In the confusion phase of the proposed encryption scheme, the input image is first separated into its three-color channels: red, green, and blue. The following procedure is then independently applied to each channel to obscure the spatial relationships between pixels, resulting in a visually unrecognizable output. To achieve this, a two-dimensional chaotic sequence matrix is generated using the proposed chaotic system. This matrix is then divided into two auxiliary index matrices: A and B, which are derived by sorting the chaotic matrix row-wise and column-wise, respectively. These matrices serve as position guides for pixel shuffling. The confusion process proceeds in two main steps:

1. Each row of the original plain image matrix **I** is rearranged based on the sorting indices from matrix A, producing an intermediate matrix Q.
2. Each column of Q is then permuted according to the column-wise sorting indices from matrix B, resulting in the scrambled matrix S.

This two-step pixel permutation is repeated for three full iterations to enhance the level of confusion. The resulting output matrix after three rounds, denoted as Ic, represents the confused version of the original image channel. This process effectively breaks the correlation between neighboring pixels and diffuses the original image structure. Figure 9 demonstrates this procedure and the detailed steps. The algorithm of the confusion phase is given in Figure 10. Due to the sensitivity of the underlying chaotic sequences to the secret key, accurate recovery of the original pixel positions is only possible with the correct key parameters.

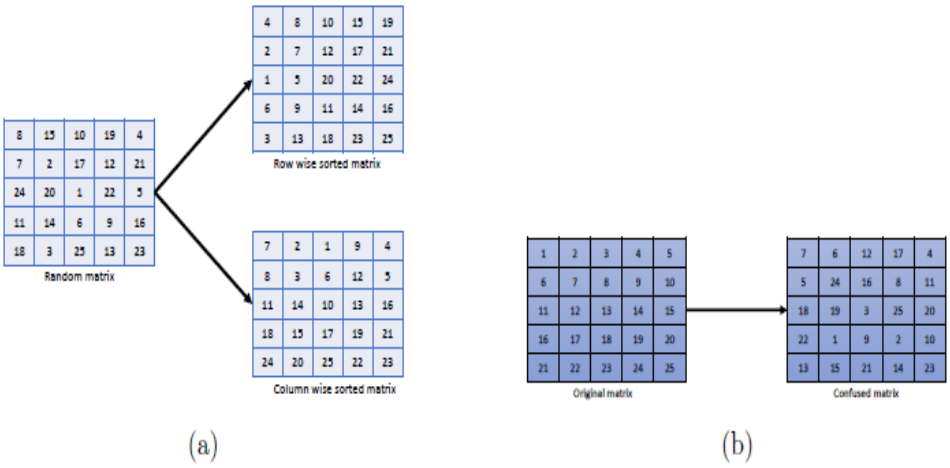


Figure 9. An example of the confusion process.

Algorithm 1: Confusion Phase Using Chaotic Sorting for a Single Color Channel

Input : An 8-bit image channel matrix $\mathcal{I} \in \mathbb{Z}^{M\times N}$, and a chaotic sequence matrix $\mathcal{X} \in \mathbb{R}^{M\times N}$

Output: The scrambled image matrix $\mathcal{I}_c$

Compute row-wise sorting indices of $\mathcal{X}$ and store in matrix $\mathcal{A}$;
Compute column-wise sorting indices of $\mathcal{X}$ and store in matrix $\mathcal{B}$;
for $i \leftarrow 1$ to M do
    for $j \leftarrow 1$ to N do
        $\mathcal{Q}(i,j) \leftarrow \mathcal{I}(\mathcal{A}(i,\mathcal{B}(i,j)),\mathcal{B}(i,j))$;
    end
end
for $j \leftarrow 1$ to N do
    for $i \leftarrow 1$ to M do
        $\mathcal{I}_c(i,j) \leftarrow \mathcal{Q}(\mathcal{A}(i,j),\mathcal{B}(\mathcal{A}(i,j),j))$;
    end
end

Figure 10. Confusion Phase Using Chaotic Sorting for a Single-Color Channel

4.3. Diffusion Process

In image encryption, the diffusion phase plays a pivotal role in enhancing the security of the encrypted image. The purpose of diffusion is to obscure the relationship between the plaintext and the ciphertext, ensuring that small changes in the input lead to significant, unpredictable changes in the output. This makes it harder for attackers to identify patterns or correlations between the original and encrypted images. Additionally, a well-designed diffusion process makes the encryption resistant to differential attacks, where small changes in the plaintext could expose the key. In this scheme, we generate a chaotic sequence matrix $\mathcal{R}$ using the 2D Hybrid Hyperchaotic Modified Lemniscate (2D-HHML) map. This matrix is of the same size as the image, and three such matrices are created for each color channel to perform the diffusion operation. Since the chaotic sequence values are bounded between 0 and 1, the entries in the matrix $\mathcal{R}$ lie within the range $\mathcal{R}_{i,j} \in (0,1)$. For an 8-bit image, where the pixel values range from 0 to 255, the values in $\mathcal{R}$ are transformed into integer values between 0 and 255, resulting in a new matrix V. The conversion is done using equation 8:

$$\mathcal{V}_{i,j} = \left(\left\lfloor \mathcal{R}_{i,j} \times 10^4 \right\rfloor\right) mod 256 \tag{8}$$

where ⌊x⌋ denotes the floor function. After this step, we perform the diffusion process as described by equation 9, where the pixel values in the image are altered based on neighboring pixel values and the matrix V:

$$\mathcal{F}_{i,j} = \begin{cases} \mathcal{P}_{i,j} \oplus \mathcal{V}_{i,j} & \text{if } i = 1 \text{ and } j = 1 \\ \mathcal{P}_{i,j} \oplus \mathcal{P}_{i,j-1} \oplus \mathcal{V}_{i,j} & \text{if } i = 1 \text{ and } j > 1 \\ \mathcal{P}_{i,j} \oplus \mathcal{P}_{i-1,j} \oplus \mathcal{V}_{i,j} & \text{if } i > 1 \text{ and } j = 1 \\ \mathcal{P}_{i,j} \oplus \mathcal{P}_{i-1,j-1} \oplus \mathcal{P}_{i-1,j} \oplus \mathcal{P}_{i,j-1} \oplus \mathcal{V}_{i,j} & \text{if } i > 1 \text{ and } j > 1 \end{cases} \tag{9}$$

where $\bigoplus$ denotes the XOR operation, and P refers to the pixel values from the previous stage (confused image). After this step, the pixel values are thoroughly mixed, resulting in the final encrypted image matrix F. The algorithm of the diffusion process is given in Figure 11. For decryption, the inverse of the diffusion process is applied, as described in equation 10:

$$\mathcal{P}_{i,j} = \begin{cases} \mathcal{F}_{i,j} \oplus \mathcal{V}_{i,j} & \text{if } i = 1 \text{ and } j = 1 \\ \mathcal{F}_{i,j} \oplus \mathcal{F}_{i,j-1} \oplus \mathcal{V}_{i,j} & \text{if } i = 1 \text{ and } j > 1 \\ \mathcal{F}_{i,j} \oplus \mathcal{F}_{i-1,j} \oplus \mathcal{V}_{i,j} & \text{if } i > 1 \text{ and } j = 1 \\ \mathcal{F}_{i,j} \oplus \mathcal{F}_{i-1,j-1} \oplus \mathcal{F}_{i-1,j} \oplus \mathcal{F}_{i,j-1} \oplus \mathcal{V}_{i,j} & \text{if } i > 1 \text{ and } j > 1 \end{cases} \tag{10}$$

Algorithm 2: Diffusion Phase Based on Modified 2D Neighboring Pixels

Input : Matrix $\mathcal{P}$ and matrix $\mathcal{V}$ of size $X \times Y$
Output: Diffused matrix $\mathcal{F}$

```
for i ← 1 to X do
    for j ← 1 to Y do
        if i = 1 and j = 1 then
            F(i, j) = P(i, j) ⊕ V(i, j)
        else
            if i = 1 then
                F(i, j) = P(i, j) ⊕ P(i, j − 1) ⊕ V(i, j)
            else
                if j = 1 then
                    F(i, j) = P(i, j) ⊕ P(i − 1, j) ⊕ V(i, j)
                else
                    F(i, j) = P(i, j)⊕P(i−1, j−1)⊕P(i−1, j)⊕P(i, j−1)⊕V(i, j)
                end
            end
        end
    end
end
```

Figure 11. Diffusion Phase Based on Modified 2D Neighboring Pixels.

This inverse operation recovers the original pixel values, ensuring that the encryption and decryption processes are accurately reversible. The step-by-step description of the decryption process is illustrated in Figure 12.

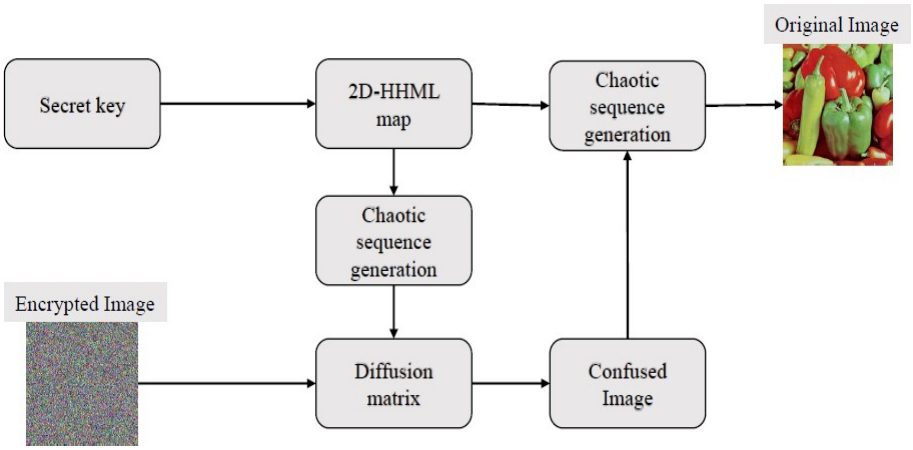


Figure 12. Flowchart of the decryption scheme.

4.4. Key Space Analysis and Key Management Security

An essential requirement of a secure cryptographic system is a sufficiently large key space that can resist brute-force attacks. In the proposed encryption scheme based on the 2D Hybrid Hyperchaotic Modified Lemniscate (2D-HHML) map, the secret key comprises three critical components: the control parameter r and the initial conditions x_0 and y_0. Each of these parameters is a real number and can be represented with a precision of at least 10^{-14}, resulting in approximately $10^{42} \approx 2^{139}$ possible combinations for the key space. If independent keys are used for each color channel (R, G, and B), the total key space expands to approximately $(2^{139})^3 = 2^{417}$, which is sufficiently large to prevent any successful brute-force attack, even when considering emerging computational threats such as quantum computing. The following aspects further strengthen the key space and address key management concerns:

- **Key Sensitivity:** The encryption scheme is highly sensitive to small changes in the values of r, x_0, or y_0. A slight variation, such as 10^{-14}, in any parameter results in a completely different ciphertext, thus ensuring resistance to differential key guessing.
- **Avoiding Weak Keys:** Certain values of the control parameter r may push the system into non-chaotic or periodic behavior. To avoid such scenarios, the Lyapunov exponent of the system should be analyzed, and only values of r leading to hyperchaotic dynamics should be selected. An automated check during key generation can help eliminate weak keys.
- **Key Management Practices:** To mitigate vulnerabilities associated with weak key selection and poor key storage, keys should be generated using cryptographically secure pseudo-random number generators (CSPRNGs) or hardware-based entropy sources. In real-world applications, secure key exchange mechanisms such as RSA, ECC, or post-quantum cryptographic protocols can be integrated to ensure safe transmission. Furthermore, keys should be stored in tamper-proof hardware modules or protected using hardware security modules (HSMs).
- **Channel-Wise Key Diversity:** Using distinct key sets for each RGB channel adds further diffusion and complexity, which enhances resistance to color-based statistical and chosen-plaintext attacks.

In summary, the encryption algorithm offers a large and robust key space and provides practical mechanisms for secure key generation, validation, and management, ensuring strong protection against brute-force and key related attacks.

5. Security analysis and Performance metrics

In this section, we assess the performance of the proposed image encryption algorithm by computing various metrics. The evaluation utilizes standard test images Lena, Baboon, and Peppers each with dimensions of 256 256 pixels. All analyses are conducted using Python 3.10 on a computer equipped with an 11th Gen Intel Core i5-1135G7 processor. Additionally, we compare the performance of our algorithm against existing encryption schemes from the literature to benchmark its effectiveness.

5.1. Visual and Histogram analyses

The primary and most crucial metric for evaluating any image encryption algorithm is its impact on the human visual system. Histogram analysis, in this context, involves examining the statistical distribution of pixel intensities in an image before and after encryption. This analysis reveals key characteristics, such as the frequency of pixel values, their distribution patterns, and intensity variations. A robust encryption algorithm should produce a nearly uniform or random-looking histogram, ensuring that the original image's pixel distribution is effectively concealed, leaving no discernible patterns or structural clues.

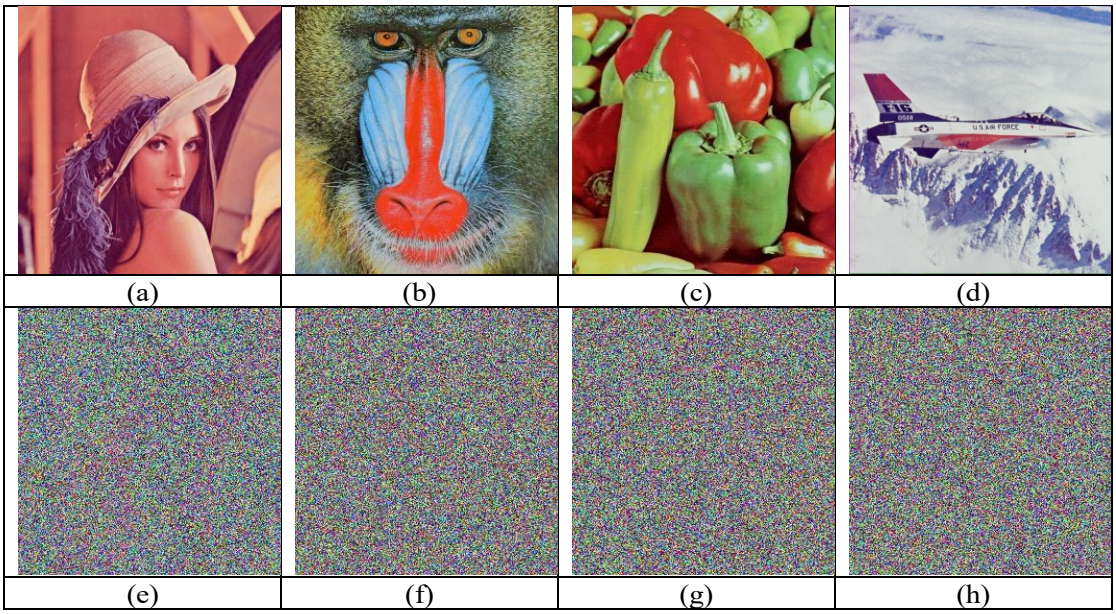


Figure 13. The plain images of (a) Lena (b) Baboon (c) Peppers (d) Airplane and encrypted images of (e) Lena (f) Baboon (g) Peppers (h) Airplane.

From the encrypted images in Figure 13 it is clear that no information about the original image can be retrieved visually.

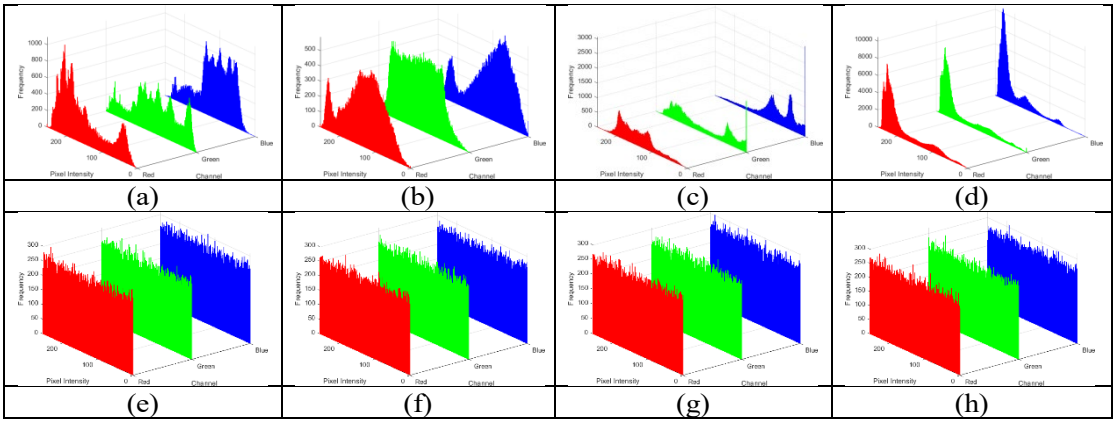


Figure 14. The histograms of plain (a) Lena (b) Baboon (c) Peppers (d) Airplane and the histograms of encrypted (e) Lena (f) Baboon (g) Peppers (h) Airplane.

Based on the histograms of the encrypted images in Figure 14 it is apparent that no statistical data pertaining to the plain image can be discerned from the encrypted image.

5.2. Information Entropy

Entropy is a critical measure used to evaluate the randomness and uncertainty present in the encrypted image. It quantifies the amount of disorder, complexity, or unpredictability within the pixel values of the encrypted image. An encrypted image's entropy illustrates how successfully the encryption technique distributed the information throughout the image, making it difficult for an unauthorised entity to

decipher the original material. A high entropy indicates a more equally dispersed as well as complex encrypted image, making significant information extraction impossible without a valid decryption key. The entropy E of an image is given by,

$$E = -\sum_{i=0}^{L-1} p(i) \log_2\big(p(i)\big)$$

where,

- L is the number of possible pixel intensities
- $p(i)$ is the probability of occurrence of the intensity level 'i' in the encrypted image.

Increasing the entropy of encrypted images is a critical goal in image encryption for ensuring robust security and confidentiality. The experimental results on the entropy of our encryption scheme is given in Table 1.

Table 1. Entropy of the encrypted images.

Image	Channels	Entropy
Lena	Red	7.9994
	Green	7.9993
	Blue	7.9994
Baboon	Red	7.9993
	Green	7.9992
	Blue	7.9994
Peppers	Red	7.9995
	Green	7.9993
	Blue	7.9993
Airplane	Red	7.9992
	Green	7.9991
	Blue	7.9992

In Table 2 we compare our results with some of the existing image encryption techniques. For this comparison, we use the same Lena image as the test image and calculate the entropy values in all three channels.

Table 2. Comparison of entropy of Peppers image with some existing algorithms.

Technique	Red	Green	Blue
[28]	7.9948	7.9958	79950
[29]	7.9991	7.9954	7.9963
[30]	7.9994	7.9994	7.9993
[31]	7.9973	7.9972	7.9975
[32]	7.9993	7.9994	79991
[33]	7.9912	7.9913	7.9914
[34]	7.9992	7.9991	7.9993
[35]	7.9992	7.9993	7.9994
[36]	7.9917	7.9914	7.9915
[37]	7.9995	7.9994	7.9995
Our proposed algorithm	7.9994	7.9993	7.9994

From the comparison table, it is clear that our scheme outperforms some existing algorithms in terms of entropy analysis.

5.3. Mean Square Error

Mean Square Error (MSE) is a metric commonly used to measure the difference between two images [38]. In the context of image encryption, MSE can be employed to quantify the distortion or error between the original image and the encrypted image. The Mean Square Error (MSE) is calculated by squaring the differences between corresponding pixels in the original image $P\ (i,j)$ and the encrypted image $E(i,j)$

across all $M \times N$ pixels. This squared error sum is then divided by the total number of pixels to determine the MSE. In the context of MSE, higher values signify improved resistance of an encryption algorithm against statistical attacks.

$$\mathrm{MSE}(P,E) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\big(P(i,j) - E(i,j)\big)^2$$

Table 3 shows the Mean Square Error of some test images in our proposed scheme.

Table 3. Mean Square Error of our technique.

Image	MSE
Lena	1672.497
Baboon	1621.383
Peppers	1788.225
Airplane	1827.987

5.4. NIST SP 800-22 Test

The National Institute of Standards and Technology Special Publication $800-22$ (NIST SP 800-22) test suite is a professional tool designed to rigorously assess the randomness of sequences across multiple criteria [39]. It offers a comprehensive evaluation from 15 distinct perspectives. To evaluate the randomness of our data, we randomly generated a 50 MB binary sequence, which we then segmented into 50 groups for thorough analysis. Detailed in Table 4, our findings reveal that all sequences achieved nearly perfect pass rates, surpassing their respective thresholds. These results underscore the capability of the constructed 2D-HHML map to reliably generate Pseudo-Random Sequences, that meet the stringent demands of practical applications.

Table 4. NIST test analysis.

Sl. No.	Test	Result (P-value)	Result
1	Frequency	1	Pass
2	Block frequency	0.461	Pass
3	Runs	0.598	Pass
4	Longest run	0.282	Pass
5	FFT	0.939	Pass
6	Rank	0.859	Pass
7	Overlapping template	0.621	Pass
8	Non-Overlapping template	0.659	Pass
9	Universal	0.789	Pass
10	Approximate entropy	0.668	Pass
11	Random Excursions	0.652	Pass
12	Random Excursions variant	0.695	Pass
13	Linear complexity	0.865	Pass
14	Serial	0.756	Pass
15	Cumulative sums	0.950	Pass

5.5. Sample Entropy

Sample entropy (SampEn) serves as a statistical measure to assess the intricacy of self-similarity within a dynamic system's time series [40]. It quantifies the degree of self-similarity inherent in the system's dynamics. To calculate the sample entropy, the 2D image was converted into a 1D time series by flattening the pixel values row by row or column by column. The parameters m, the embedding dimension, and r, the tolerance level, were defined. Next, vectors of length m were constructed from the time series, denoted as $\{X_i = (x_i, x_{i+1}, \dots, x_{i+m-1})\}$ For each vector X_i, the distance to all other vectors X_j was computed using the maximum norm. The number of vectors X_j that were within a distance r of X_i was counted, and this count was denoted as B_i. These counts were normalized by the total number of vectors, $N - m + 1$, to determine the probability of similarity. This process was repeated for vectors of length $m + 1$, counting the number of similar vectors, denoted as A_i. Finally, the sample entropy

was computed using,

$$\text{SampEn}(m, r) = -\ln\left(\frac{\sum A_i}{\sum B_i}\right),$$

where $\sum A_i$ and $\sum B_i$ were the sums of counts obtained from the steps described above. This method provided a quantifiable measure of the complexity and unpredictability of the encrypted image. An increased Sample entropy indicates reduced regularity and, consequently, higher randomness within the dynamic system. Table 5 provides a summary of their respective sample entropy values. For the proposed chaotic system, the sample entropy is 2.231, while the best among the others is 2.1557 [41]. These findings demonstrate that our fractal chaotic map generates time series of greater complexity compared to some previously proposed maps. Figure 15 shows the comparison of Sample entropy of some well known maps with our proposed 2D-HHML map.

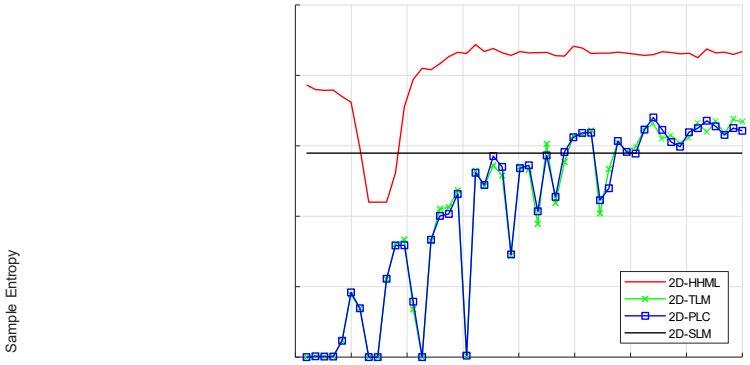


Figure 15. Comparison of sample entropy of some maps.

Table 5. Comaprision of Lyapunov exponent and Sample entropy.

Chaotic maps	Lyapunov exponent	Sample entropy
[42]	5.596	0.339
[43]	2.749	1.766
[44]	2.066	1.764
[45]	4.514	1.937
[41]	10.678	2.155
Proposed map	17.350	2.231

5.6. Correlation coefficient analysis

Correlation coefficient analysis in the context of image encryption typically refers to the examination of statistical relationships between pixels or image components before and after encryption [46]. The correlation coefficient is a measure of the strength and direction of a linear relationship between two variables. In image encryption, it is used to assess the degree of correlation or similarity between the original and encrypted images. The correlation coefficient (rxy) is given by:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x) \cdot D(y)}} \tag{11}$$

where $cov(x, y)$ is the covariance between x and y, defined as:

$$\text{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{12}$$

$D(x)$ is the variance of x, defined as:

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{13}$$

$D(y)$ is the variance of y, defined as:

$$(y) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{14}$$

and $E(x)$ is the mean (expected value) of x, defined as:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{15}$$

To enhance resilience against statistical attacks, an encryption scheme should exhibit a correlation coefficient near zero for adjacent pixels in all three directions: horizontally (H), vertically (V), and diagonally (D). A correlation coefficient close to zero signifies a lack of correlation, while values nearing -1 indicate strong negative correlation and values approaching 1 signify strong positive correlation. Encryption schemes with correlation coefficients deviating significantly from zero make it susceptible to exposure through statistical attacks on the encrypted image.

Table 6. Correlation coefficient of the plain images.

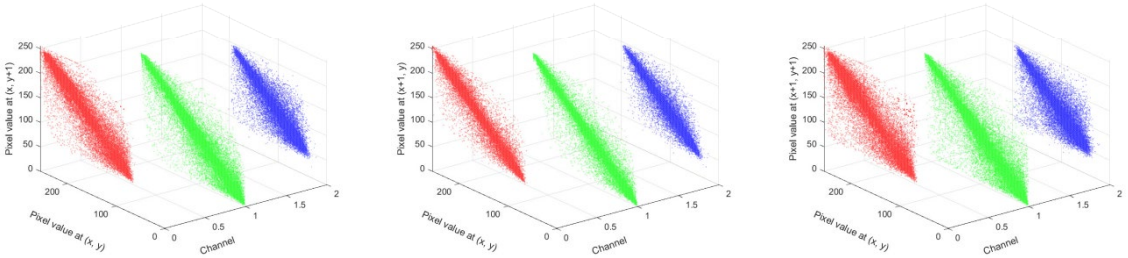
Image	Channels	Horizontal	Vertical	Diagonal
Lena	Red	0.94787	0.97340	0.92594
	Green	0.94803	0.97346	0.92709
	Blue	0.90954	0.94968	0.87975
Baboon	Red	0.91016	0.87957	0.86476
	Green	0.86124	0.81778	0.79248
	Blue	0.91673	0.89412	0.87577
Peppers	Red	0.95997	0.96444	0.92976
	Green	0.97867	0.98267	0.96260
	Blue	0.95560	0.96270	0.92375

From Table 6, it can be seen that the all the channels of the plain images are highly correlated in horizontal, vertical and diagonal directions. Similarly, Table 7 shows that the channels of the encrypted images are not correlated in horizontal, vertical and diagonal directions.

Table 7. Correlation coefficient of the encrypted images.

Image	Channels	Horizontal	Vertical	Diagonal
Lena	Red	0.08341	0.08075	0.07364
	Green	0.08049	0.08173	0.00690
	Blue	0.06913	0.06717	0.06253
Baboon	Red	0.06605	0.06564	0.06082
	Green	0.07994	0.07893	0.06510
	Blue	0.06808	0.06328	0.06015
Peppers	Red	0.07619	0.06654	0.06712
	Green	0.08747	0.07871	0.07510
	Blue	0.07046	0.05978	0.06374

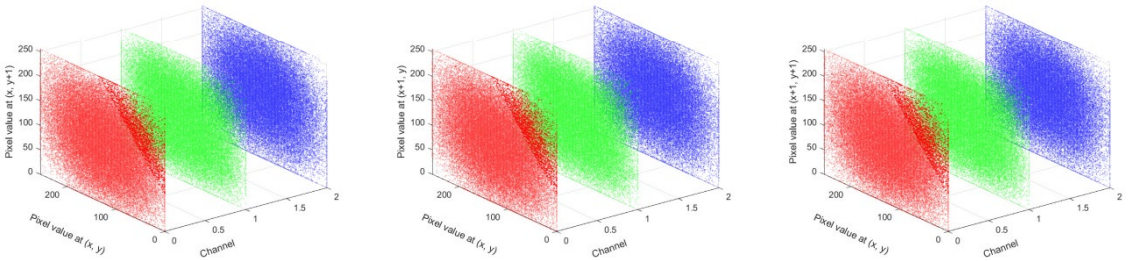
The discrepancies in values between Table 6 and Table 7 can be visually verified by examining the following images. It provides insights into how well the algorithm obscures the inherent patterns in the original image, contributing to the overall security of the encryption scheme.



(a) (b) (c)

Figure 16. The correlation coefficient of Plain Lena image.

Figure 16 shows the correlation coefficient of plain Lena image in Horizontal, Vertical and Diagonal directions. Also, Figure 17 shows the correlation coefficient of encrypted Lena image in Horizontal, Vertical and Diagonal directions. From these two images it is evident that the dependency of adjacent pixels has reduced drastically in horizontal, vertical and diagonal directions.



(a) (b) (c)

Figure 17. The correlation coefficient of encrypted image.

5.7. Differential attack analysis

One of the essential security properties of a robust image encryption scheme is its sensitivity to slight changes in the input [47]. A well-designed cryptosystem should exhibit the avalanche effect, wherein a minimal modification (such as altering a single pixel or a single bit of the plain image or key) results in a significantly different encrypted image. **Number of Pixel Change Rate (NPCR)** and **Unified Average Changing Intensity (UACI)** are two essential metrics for assessing the resistance of image encryption schemes against differential attacks.

5.7.1. Number of Pixel Change Rate (NPCR)

NPCR measures the percentage of pixels that change in the cipher image when a single pixel in the plain image is modified. It is defined as:

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%$$

where $D(i,j) = 1$ if the pixel at position (i,j) in the two cipher images differs, and $D(i,j) = 0$ otherwise, and $M \times N$ is the image size. A high NPCR (close to 100%) indicates that the encryption scheme is highly sensitive to small changes in the input, a key requirement to thwart differential attacks. In our scheme, the observed NPCR exceeds 99.6%, which demonstrates excellent pixel-level diffusion and ensures that any minor change in the plaintext results in a dramatically different ciphertext.

5.7.2. Unified Average Changing Intensity (UACI)

UACI measures the average intensity of differences between two cipher images corresponding to slightly different plaintexts. It is calculated as:

$$\text{UACI} = \frac{1}{M \times N} \sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100$$

where $C_1(i,j)$ and $C_2(i,j)$ are corresponding pixel values in two cipher images. UACI reflects the strength of intensity-level diffusion across the image. A UACI value near 33% is considered optimal for 8-bit grayscale images. The proposed scheme achieves an average UACI of 33.51%, confirming effective pixel intensity obfuscation. Table 8 shows the NPCR and UACI values of our proposed method.

5.7.3. Correlation with Practical Usability

The high NPCR and optimal UACI values ensure that:

- **Security:** The encryption scheme is highly resilient to differential attacks such as chosen-plaintext or known-plaintext attacks.
- **Unpredictability:** Even minimal changes in input (e.g., a single pixel or bit) result in a completely different encrypted image, ensuring non-repeatability and robustness in dynamic environments.
- **Suitability for Real-World Applications:** In practical systems such as secure medical imaging or surveillance feeds, where image frames may be similar, high NPCR and UACI ensure that the attacker cannot infer any structure or patterns across frames.

Thus, NPCR and UACI are not only theoretical indicators of differential resistance but also critical for ensuring privacy in practical deployments where slight variations in consecutive frames or data may exist.

Table 8. NPCR and UACI values for the RGB channels of the images.

Image	Channels	NPCR	UACI
Lena	Red	0.9960	0.3340
	Green	0.9962	0.3342
	Blue	0.9961	0.3339
Baboon	Red	0.9962	0.3341
	Green	0.9959	0.3338
	Blue	0.9959	0.3340
Peppers	Red	0.9960	0.3339
	Green	0.9961	0.3340
	Blue	0.9959	0.3337
Airplane	Red	0.9963	0.3410
	Green	0.9958	0.3344
	Blue	0.9965	0.3339

5.8. Chosen-Plaintext Attack

The chosen-plaintext attack is commonly used to decipher ciphertext images, enabling an attacker to gather insights about the encryption key or system by examining plaintext messages alongside their corresponding ciphertexts [48]. In this study, we assess the encryption system's resilience against chosen-plaintext attacks using the following formula [49]:

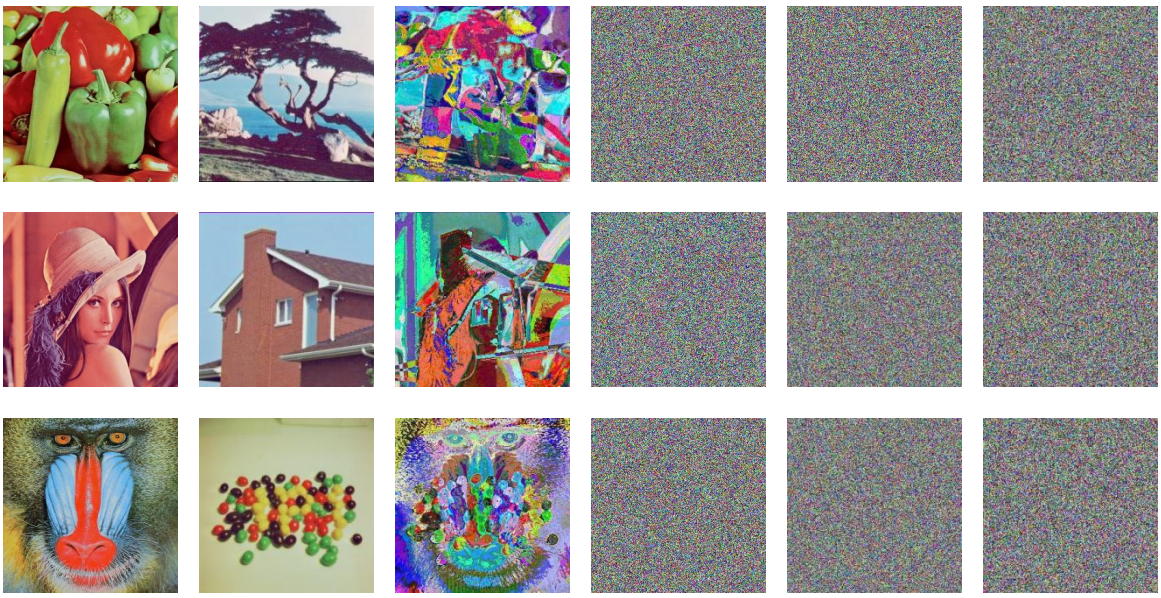
$$I1(i,j) \oplus I2(i,j) \neq C1(i,j) \oplus C2(i,j)$$

where I1 and I2 represent the plaintext images, and C1 and C2 are the corresponding ciphertext images. The higher the number of pixels that satisfy this inequality, the more resistant the algorithm is to chosen-plaintext attacks. Figure 18 demonstrates the test results for images of various sizes, while Table 9 shows the total number of pixels meeting the inequality and their respective percentages. The experimental

outcomes indicate that the proposed algorithm exhibits strong resistance to chosen-plaintext attacks.

Table 9. Test results of chosen plaintext attack.

Image 1	Image 2	Total number	Percentage
Peppers	Tree	65499	99.91
Lena	House	65472	99.90
Baboon	Jelly	65469	99.89



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

Figure 18. Chosen plaintext attack test: (a) Plain images I_1; (b) Plain images I_2; (c) $I_1 \oplus I_2$; (d) Encrypted images C_1 of I_1; (e) Encrypted images C_2 of I_2; (f) $C_1 \oplus C_2$.

5.9. Structural Similarity Index Measure

Structural Similarity Index Measure (SSIM) is a metric that quantifies the similarity between two images by considering three components: luminance, contrast, and structure. The SSIM index ranges from $-1\ to\ 1$, where 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates complete dissimilarity. If X and Y are the two images then the Structural Similarity Index Measure is given by:

$$\text{SSIM}(X,Y) = [l(X,Y) \cdot c(X,Y) \cdot s(X,Y)]^{\alpha}$$

where,

$$l(X,Y) = \frac{2\mu_X\mu_Y + C_1}{\mu_X^2 + \mu_Y^2 + C_1}$$

$$c(X,Y) = \frac{2\sigma_X\sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2}$$

$$s(X,Y) = \frac{\sigma_{XY} + C_3}{\sigma_X\sigma_Y + C_3}$$

where,

μ_X and μ_Y are the average pixel intensities of images X and Y, respectively.
σ_X and σ_Y are the standard deviations of pixel intensities of images X and Y, respectively.
σ_{XY} is the covariance of pixel intensities between images X and Y.
C_1, C_2, and C_3 are small constants to prevent division by zero and numerical instability.
These formulas incorporate the mean (μ), standard deviation (σ), and covariance (σ_{XY}) of pixel intensities in the calculation of luminance, contrast and structure respectively.

Table 10. SSIM of original image and encrypted image.

Plain Image	Encrypted Image	Decrypted Image
Airplane	0.021	1
Lena	0.012	1
Baboon	-0.033	1
Peppers	0.029	1

Table 10 shows the SSIM of the original image with the encrypted image and original image with the decrypted image. From this table it is clear that our encrypted images have no similarity with the original images and our decrypted images have perfect similarity with the original images.

5.10. Noise and Data Loss Analysis

During the transmission of images, communication channels are often subject to various forms of interference, which can lead to data loss or contamination by noise. Consequently, it is crucial for the encryption algorithm to exhibit robust resilience to both noise and cropping attacks. This ensures that the quality and decryption capability of the encrypted image are preserved, even when faced with noise or incomplete data [50]. In our study, we investigated the effects of cropping the cipher image by different fractions, specifically 6.25%, 12.5%, 18.75%, and 25%, and assessed the restoration capabilities, as depicted in Figure 19. Also, we specifically examine the impact of salt and pepper (SP) noise on the encrypted image [51]. As illustrated in Figure 20, we introduced SP noise at varying densities of 1%, 5%, 20%, and 40% to the cipher image. Following this, we decrypted the image to evaluate the algorithm's performance in handling noise-induced distortions. The results from these experiments demonstrate that our proposed algorithm maintains a high level of resistance to both data loss and noise contamination. The decrypted images retained their quality and intelligibility, even under significant noise levels and partial data loss due to cropping. This robustness is indicative of the algorithm's effectiveness in ensuring reliable image transmission and decryption in adverse conditions, thereby making it a viable solution for secure image communication.

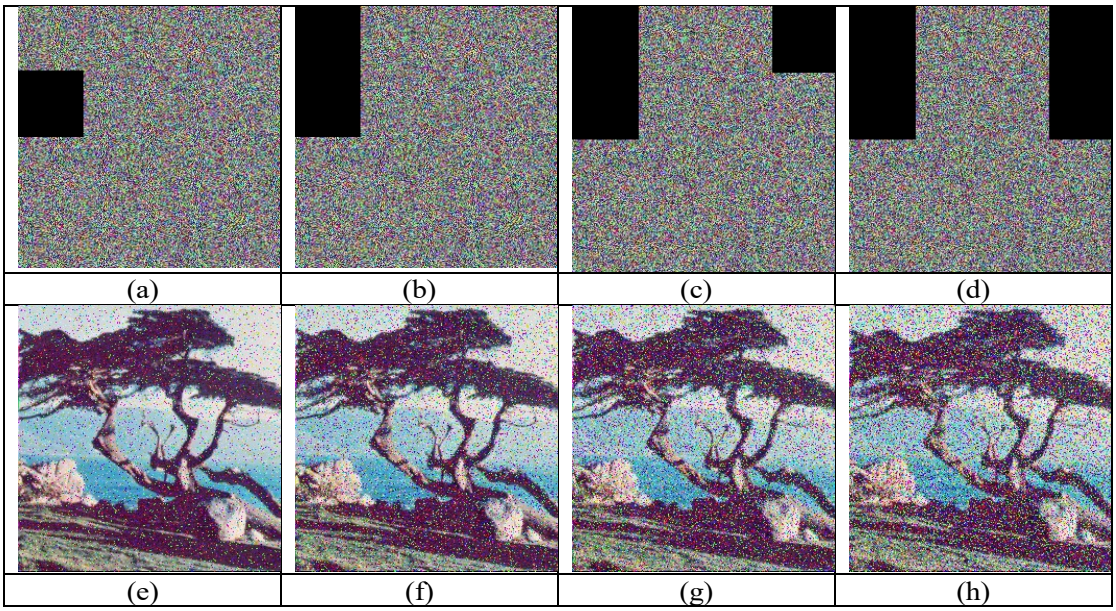


Figure 19. Data loss analysis of encrypted images with (a) 6.25% (b) 12.5% (c) 18.75% (d) 25% data loss, (e), (f), (g) and (h) represents their corresponding decrypted images.

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

Figure 20. Decryption with Salt and Pepper noise (a) 1% (b) 5% (c) 20% (d) 40%.

5.11. Computational Complexity and Memory Usage Analysis

In practical deployments, especially on mobile and embedded systems with limited resources, it is crucial to assess the computational and memory overhead introduced by the encryption algorithm. This section presents an analysis of the proposed 2D-HHML based image encryption scheme in terms of time complexity, computational cost, and memory requirements. The encryption process involves the following major steps:

- Generation of chaotic sequences using the 2D-HHML map.
- Pixel-level permutation (confusion phase).
- Pixel-level diffusion using the generated sequences.

Assuming an input image of size $M \times N$ with $L = M \times N$ pixels:

- Chaotic sequence generation has a linear complexity of $O(L)$.
- Permutation of pixel positions also requires $O(L)$ time.
- Diffusion involves operations such as addition and XOR, which are $O(1)$ per pixel, leading to $O(L)$ complexity.

Total Time Complexity:

$$T(L) = O(L) + O(L) + O(L) = O(L)$$

This linear complexity ensures scalability and efficiency for images of varying resolutions. To empirically evaluate the runtime performance, the encryption algorithm was implemented in Python 3.10 and tested on different platforms:

- **PC Platform:** Intel i7 @ 2.60GHz, 16GB RAM
- **Mobile Platform:** ARM Cortex-A73 @ 2.2GHz, 4GB RAM (via Pydroid)

The execution time for a 256×256 image is given in Table 11.

Table 11. Execution Time for 256×256 Image.

Platform	Encryption Time	Decryption Time
PC (Python)	0.095 s	0.098 s
Mobile (ARM, Python)	0.423 s	0.437 s

These results demonstrate the algorithm's suitability for both desktop and constrained environments with minor trade-offs in execution time on mobile platforms.

5.12. Memory Usage Analysis

The primary memory requirements arise from:

- Storage of the original image (L bytes for grayscale, $3L$ bytes for RGB).
- Temporary arrays for chaotic sequences: 2 sequences of L floating-point values ($2L \times 4$ bytes assuming 32-bit float).
- Additional buffer for storing the encrypted image.

Total Memory Footprint (Grayscale Image):

$Memory = L + (2L \times 4) + L = 10L$ bytes

For a 256×256 grayscale image:

$$L = 65536 \Rightarrow Memory \approx 640\ kB$$

Even for RGB images, the memory usage remains under 2 MB, making the algorithm feasible for embedded or IoT systems with moderate RAM capacities.

5.13. Optimization Potential for Embedded Platforms

To further optimize the algorithm for embedded systems:

- Replace Python with C or C++ for faster execution.
- Use fixed-point arithmetic instead of floating-point where feasible.
- Generate chaotic sequences on-the-fly to avoid storing large arrays.
- Use memory-efficient data structures (e.g., in-place transformations).

Thus the proposed algorithm demonstrates low computational complexity, modest memory requirements, and competitive execution time, making it highly suitable for secure applications on mobile devices, IoT systems, and real-time embedded platforms.

5.14. Hardware Implementation and Applications

The proposed 2D Hybrid Hyperchaotic Modified Lemniscate Map (2D-HHML) based image encryption algorithm can be effectively implemented on hardware platforms such as Field Programmable Gate Arrays (FPGAs) and advanced microcontrollers. Due to the deterministic structure and simple mathematical operations (involving trigonometric and polynomial expressions), the algorithm is well-suited for real-time hardware execution.

FPGA Implementation: The encryption system can be structured using hardware description languages (HDL) such as Verilog or VHDL. Trigonometric operations are optimized using Look-Up Tables (LUTs), while fixed-point arithmetic reduces resource consumption. Platforms like Xilinx Artix-7 or Zynq-7000 can execute real-time encryption on image frames with high throughput and low latency.

Microcontroller Implementation: The algorithm is compatible with microcontrollers (e.g., ARM Cortex-M4/M7) equipped with DSP and floating-point units. Leveraging CMSIS-DSP libraries and optimizing iterations enable efficient static image encryption

for edge computing and IoT applications. Security measures such as constant-time execution and protection against side-channel attacks can be incorporated to harden the hardware implementation. The proposed encryption scheme has wide applicability in various domains, including:

- **Medical Imaging:** Ensures secure transmission and storage of MRI, CT, and X-ray images in telemedicine and hospital systems.
- **Military Surveillance:** Secures confidential reconnaissance and drone imaging data in real-time.
- **Cloud Security:** Facilitates encrypted image uploads to cloud servers to prevent data breaches and unauthorized access.
- **IoT and Edge Devices:** Suitable for lightweight encryption in resource-constrained environments such as smart homes, UAVs, and portable devices.
- **Copyright Protection:** Prevents illegal redistribution and forgery of digital multimedia content.

The high efficiency, low complexity, and strong security of the scheme make it suitable for both high-performance and constrained environments.

5.15. Scalability Analysis

Scalability is a key requirement for any encryption algorithm intended for real-world applications, particularly in domains where images of different resolutions and complexities must be handled efficiently. These include applications such as medical imaging, satellite data transmission, real-time surveillance, and mobile multimedia protection. This section evaluates how the proposed 2D Hybrid Hyperchaotic Modified Lemniscate Map (2D-HHML)-based image encryption scheme performs when the image size and content complexity vary. The focus is on execution time, memory usage, and the algorithm's adaptability to images with differing statistical and structural characteristics. The encryption algorithm was implemented in Python and tested on a standard personal computer equipped with an Intel Core i7 processor operating at 2.60 GHz and 16 GB RAM. For this evaluation, test images of sizes 128 × 128, 256 × 256, 512 × 512, and 1024 × 1024 were selected to observe how the algorithm scales with increasing pixel dimensions. The primary metrics considered are the time taken for encryption and the peak memory required during the process. Table 12 gives the encryption time and memory usage for images with different resolutions.

Table 12. Encryption Time and Memory Usage for Different Image Resolutions.

Image Size	Pixel Count (L)	Encryption Time (s)	Memory Usage (MB)
128×128	16,384	0.021	0.16
256×256	65,536	0.095	0.63
512×512	262,144	0.39	2.50

1024×1024	1,048,576	1.59	10.0

As observed in Table 1, the encryption time and memory usage both exhibit an approximately linear increase with respect to the number of image pixels. This aligns with the theoretical time complexity of the algorithm, which is $O(L)$, where $L = M \times N$ is the total number of pixels in the image. Each phase of the encryption process including chaotic sequence generation, permutation (confusion), and pixel-wise diffusion operates linearly with the image size, making the algorithm efficient even for high resolution images. While image size determines the volume of data processed, image complexity measured in terms of texture, entropy, and pixel intensity variation can influence the runtime of certain image processing algorithms. To verify whether the encryption performance is affected by content complexity, we selected three grayscale images of size 256 × 256 representing low, medium, and high complexity:

- **Low Complexity Image:** A smooth image with large uniform regions and low entropy.
- **Medium Complexity Image:** A natural scene with moderate texture and details.
- **High Complexity Image:** An image filled with noise or high-frequency details, exhibiting high entropy.

Table 13 illustrates the execution time vs. image complexity for 256×256 images.

Table 13. Execution Time vs. Image Complexity (256×256 images).

Complexity Level	Shannon Entropy	Encryption Time (s)
Low Detail	5.2	0.093
Medium Detail	7.1	0.095
High Detail	7.9	0.096

The results confirm that the encryption time remains effectively constant across different levels of image complexity. This behavior arises because the chaotic key stream generation and pixel operations (XOR, addition, modulo) are executed uniformly across all pixels, independent of the image's pixel intensity distribution or structural details. This content-agnostic behavior ensures predictable performance in dynamic and heterogeneous image environments.

5.16. Practical Implications

The scalability tests indicate the following key strengths of the proposed encryption algorithm:

- **Linear Growth with Image Size:** The algorithm scales linearly in both time and memory as the image resolution increases, validating its suitability for processing high-resolution data in domains such as medical imaging (e.g., DICOM images), aerial surveillance, and satellite imagery.
- **Independence from Image Complexity:** The runtime is consistent across images with different levels of texture and entropy. This implies that the encryption overhead remains predictable and reliable even in video encryption tasks where scene complexity may vary frame-by-frame.
- **Efficient Memory Utilization:** With memory usage remaining within 10 MB for 1-megapixel grayscale images (or 30 MB for RGB images), the algorithm is lightweight and can be ported to resource-constrained platforms such as Raspberry Pi, STM32 microcontrollers, or edge IoT devices.
- **Real-Time Potential:** Even with a non-optimized Python implementation, the encryption of a 256 × 256 image is completed in less than 0.1 seconds, demonstrating promise for real-time applications. Further optimization using C/C++ or parallel processing techniques would make it viable for streaming and embedded use cases.

The proposed image encryption algorithm based on the 2D-HHML exhibits excellent scalability properties. It handles increasing image sizes with linear time and memory growth and remains unaffected by content complexity. This makes the algorithm a robust and adaptable solution for a wide range of image security applications, from mobile devices to high-performance computing environments.

5.17. Comparative Analysis

To thoroughly assess the security of the proposed algorithm, we compared its performance metrics χ^2, Correlation, Information entropy (IE), NPCR, UACI and encryption time in seconds with those reported for several recent algorithms. When average values for these metrics were available in the literature, we used them directly; otherwise, we computed the mean values ourselves. The comparison results are summarized in Table 14. As evidenced by the table, the proposed algorithm achieves performance scores on par with, or superior to, existing state-of-the-art methods, thereby demonstrating

its robustness against statistical and differential attacks.

Table 14. Comparison on several performance indexes.

Algorithm	χ^2	Correlation			IE	NCPR	UACI	Time
		H	V	D				
[28]	246.4266	0.00816	0.00317	0.00323	7.99555	99.6322%	33.5355%	1.092 s
[29]	-	-0.00253	0.00522	0.00456	7.9922	99.4967%	33.4513%	1.321 s
[30]	240.2557	0.03537	0.01323	0.0092	7.99132	99.7128%	33.356%	2.232 s
[52]	-	0.00016	0.00114	0.00005	7.99685	99.61526%	33.47628%	1.132 s
[53]	-	0.00877	-0.00401	-0.00405	7.99933	99.60748%	33.45856%	0.992 s
[54]	253.12552	0.00019	0.00017	0.00023	7.99942	99.6205%	33.47389%	1.637 s
[55]	261.38145	-0.00316	-0.00401	0.00571	7.99865	99.61091%	33.67948%	1.252 s
[56]	-	-0.00017	-0.00073	0.00036	7.99862	99.59667%	33.45833%	2.012 s
[57]	235.78907	0.00367	0.00223	0.01842	7.99753	99.61728%	33.27536%	1.390 s
[58]	248.36201	0.00553	0.01083	0.00710	7.99730	99.60458%	33.47221%	0.988 s
Proposed	253.14278	0.00136	0.00121	0.00112	7.99942	99.60514%	33.45917%	0.916 s

5.18. Security Assumptions and Threat Model

To strengthen the clarity and completeness of our security analysis, this subsection explicitly defines the security assumptions and the threat model under which the proposed encryption scheme is evaluated.

5.18.1. Security Assumptions

The design of the proposed encryption algorithm is based on the following security assumptions:

- **Chaotic Map Secrecy:** The parameters and initial conditions of the 2D-HHML chaotic map used in both the scrambling and diffusion stages are assumed to be secret and are shared only through secure key exchange mechanisms.
- **Key Space and Sensitivity:** The system assumes that the key space is sufficiently large ($\geq 2^{256}$) to prevent brute-force attacks, and the encryption process is highly sensitive to small variations in initial conditions or parameters.
- **Channel Insecurity:** It is assumed that the transmission channel is insecure and that attackers may intercept encrypted images but not the secret keys or internal system states.

5.18.2. Threat Model

The proposed cryptosystem is evaluated under a well-defined threat model where the attacker is assumed to have full access to the communication channel but no access to secret keys or internal states of the system. The attacker may exploit various strategies based on available information and computational capabilities. We consider the following threat scenarios:

- **Ciphertext-Only Attacks (COA):** In this scenario, the attacker has access only to the encrypted image and attempts to extract useful information without knowing the original image or encryption parameters. Our scheme mitigates such attacks by introducing a high degree of randomness in the pixel positions and values using a hybrid hyperchaotic map. The pixel value distributions of the encrypted images are nearly uniform, and correlation among adjacent pixels is eliminated, making statistical inference practically impossible.
- **Known-Plaintext Attacks (KPA):** Here, the attacker possesses one or more pairs of plaintext images and their corresponding ciphertexts. The encryption process incorporates high sensitivity to the initial conditions and key parameters of the 2D-HHML map. Even minor variations in input or key values result in vastly different encrypted outputs due to the chaotic behavior. Therefore, the system prevents the attacker from deducing the key or reverse-engineering the encryption structure.
- **Chosen-Plaintext Attacks (CPA):** In this case, the attacker is assumed to have the capability to choose specific plaintext images and obtain their corresponding ciphertexts. This type of attack simulates adversarial settings where an attacker may try to model the system through repeated queries. However, the nonlinear diffusion process and chaotic permutation sequences, which depend on dynamically generated keys and states, introduce significant unpredictability and disrupt any potential pattern learning.
- **Differential Attacks:** These involve analyzing the impact of small changes in the plaintext on the resulting ciphertext to derive the key or mapping rules. Our scheme's dual-phase encryption with both confusion and diffusion ensures that even a one-pixel change in the input causes substantial differences in the output, measured by high NPCR and UACI values. This avalanche

like behavior renders differential attacks ineffective.

- **Statistical Attacks:** These attacks attempt to exploit statistical properties of images, such as histograms, correlations, and entropy, to uncover underlying patterns. By flattening the histogram, removing pixel correlation, and achieving near-maximal entropy values, the proposed scheme ensures statistical indistinguishability of encrypted images from random noise, defeating such forms of analysis.
- **Brute-Force Attacks:** The proposed algorithm relies on a key space significantly larger than 2^{256}, incorporating multiple chaotic parameters. This expansive space, combined with high key sensitivity and interdependency between keys and system states, renders exhaustive key search computationally infeasible.
- **Side-Channel Attacks:** The side-channel attacks exploit physical implementation leaks such as timing information, power consumption, or electromagnetic emissions. While the algorithm itself does not inherently address these, it is recommended that hardware or embedded implementations adopt standard countermeasures such as execution time equalization, power masking, and secure hardware design to mitigate such risks. Ensuring constant-time execution and shielding critical operations will enhance the resilience of the encryption scheme against these physical layer threats.

6. Conclusion and Future work

In this paper, we have introduced a robust symmetric cryptosystem for the secure transmission of RGB color images, addressing key limitations of existing chaotic map-based encryption schemes. The novel 2D Hybrid Hyperchaotic Modified Lemniscate Map (2D-HHML) offers significant improvements in both security and performance, with its exceptionally high Lyapunov exponent and remarkable sensitivity to initial conditions. Through comprehensive analysis using time series, sample entropy, and the NIST SP 800-22 test, the 2D-HHML map has demonstrated superior cryptographic strength, outperforming traditional chaotic maps in terms of randomness and unpredictability. The proposed encryption scheme, which consists of image scrambling and diffusion phases, ensures that the encrypted image is highly resistant to statistical attacks and other cryptanalytic techniques. The pixel shuffle operator in the scrambling phase disrupts pixel relationships, while the diffusion phase, guided by the chaotic map, obliterates any residual correlations between the original and encrypted images. This combination of techniques makes the system highly secure and suitable for practical applications, where maintaining image integrity is paramount. Our cryptosystem stands out for its applicability in real-time environments, offering a reliable solution for scenarios where rapid encryption and decryption are required without compromising security. Its performance makes it particularly valuable for use cases such as secure medical imaging, where the integrity of patient data is critical, cloud-based data protection, where privacy concerns are prevalent, and multimedia security, where image and video content must be safeguarded against unauthorized access. Future research directions could focus on optimizing the cryptosystem for resource-constrained platforms, such as Internet of Things (IoT) devices, by investigating hardware acceleration and software optimization techniques. Moreover, enhancing the security with adaptive key generation and integrating the scheme with image compression algorithms would improve its real-time processing efficiency. Additionally, exploring quantum-resistant methods and developing machine learning-based adaptive attacks would further strengthen the encryption scheme against emerging threats, ensuring its long-term viability in the ever-evolving landscape of cybersecurity. In conclusion, the proposed cryptosystem not only outperforms existing encryption schemes but also holds strong promise for various critical applications, with potential for further advancements that will push the boundaries of secure image transmission.

Author Contributions
J.J.: writing-original draft, conceptualization, methodology; P.R.: supervision, software, validation. All authors have read and agreed to the published version of the manuscript.

Funding
This research received no external funding.

Conflict of Interest Statement
The authors declare no conflicts of interest relevant to the content presented in this article.

Data Availability Statement
All data generated or analyzed during this study are included in this article.

References

1. Kaur, M., Kumar, V.: A comprehensive review on image encryption techniques. Arch. Comput. Methods Eng. 27(1), 15–43 (2020)
2. Parvaz, R., Khedmati, Y.Y., Behroo, Y.: A new 4D chaos system with an encryption algorithm for color and grayscale images. Int. J. Bifurcation Chaos 32(14), 2250214 (2022)
3. Alawida, M.: A novel chaos-based permutation for image encryption. J. King Saud Univ.-Comput. Inf. Sci. 35(6), 101595 (2023)
4. Zhang, J., Guo, J., Lu, D.: An efficient image encryption algorithm based on S-box and DNA code. Meas.: Sensors 29, 100894 (2023)
5. Zhang, X., Zhang, X.: Image encryption algorithm based on the Matryoshka transform and modular-inverse matrix. Nonlinear Dynam. (2023)
6. Li, S., Chen, G., Cheung, A., Bhargava, B., Lo, K.T.: On the design of perceptual MPEG-video encryption algorithms. IEEE Trans. Circuits Syst. Video Technol. 17(2), 214–223 (2007)
7. Mansouri, A., Wang, X.Y.: A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. Inf. Sci. 520, 46–62 (2020)
8. Hua, Z.Y., Zhou, Y.C., Huang, H.J.: Cosine-transform-based chaotic system for image encryption. Inf. Sci. 480, 403–419 (2019)
9. Zhang, Y.: The fast image encryption algorithm based on lifting scheme and chaos. Inf. Sci. 520, 177–194 (2020)
10. Kocak, O., Erkan, U., Toktas, A., Gao, S.: PSO-based image encryption scheme using modular integrated logistic exponential map. Expert Syst. Appl. 237, 121452 (2024)
11. Matthews, R.: On the derivation of a chaotic encryption algorithm. Cryptologia 13(1), 29–42 (1989)
12. Hénon, M.: A two-dimensional mapping with a strange attractor. In: The theory of chaotic attractors, pp. 94–102. Springer, New York, NY (1976)
13. Parvees, M.M., Samath, J.A., Raj, I.K., Bose, B.P.: A color byte scrambling technique for efficient image encryption based on combined chaotic map: Image encryption using combined chaotic map. In: 2016 International conference on electrical, electronics, and optimization techniques (ICEEOT), pp. 1067–1072. IEEE (2016)
14. Khedmati, Y., Parvaz, R., Behroo, Y.: 2D Hybrid chaos map for image security transform based on framelet and cellular automata. Inform. Sci. 512, 855–879 (2020)
15. Parvaz, R., Zarebnia, M.: A combination chaotic system and application in color image encryption. Opt. Laser Technol. 101, 30–41 (2018)
16. Tutueva, A.V., Nepomuceno, E.G., Karimov, A.I., Andreev, V.S., Butusov, D.N.: Adaptive chaotic maps and their application to pseudo-random numbers generation. Chaos Solitons Fractals 133, 109615 (2020)
17. Fridrich, J.: Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurcation Chaos 8(06), 1259–1284 (1998)
18. Zarebnia, M., Pakmanesh, H., Parvaz, R.: A fast multiple-image encryption algorithm based on hybrid chaotic systems for gray scale images. Optik 179, 761–773 (2019)
19. Li, S., Chen, G., Mou, X.: On the dynamical degradation of digital piecewise linear chaotic maps. Int. J. Bifurcation Chaos 15(10), 3119–3151 (2010)
20. Jackson, J., and R. Perumal.: A Medical Image Encryption Technique Using Tropical Semiring. In International Conference on Soft Computing and Pattern Recognition, pp. 15-24. Cham: Springer Nature Switzerland, (2023)
21. Jackson, J., and R. Perumal.: A robust image encryption technique based on an improved fractional order chaotic map. Nonlinear Dynamics, 113(7): 7277-7296 (2025)
22. Jackson, J., and R. Perumal.: A novel 2D Hyperchaotic Sine Logistic map based image encryption scheme. Journal of Optics: 1-16 (2024)
23. Ponmaheshkumar, A., and R. Perumal.: A one-dimensional cosine-arcsine chaotic map for image encryption. Journal of Optics: 1-20. (2025)
24. Cao, Lv-Chen, Yu-Ling Luo, Sen-Hui Qiu, and Jun-Xiu Liu.: A perturbation method to the tent map based on Lyapunov exponent and its application. Chinese Physics B 24, no. 10: 100501 (2015)
25. Li, Chengqing, Tao Xie, Qi Liu, and Ge Cheng.: Cryptanalyzing image encryption using chaotic logistic map. Nonlinear dynamics 78: 1545-1551 (2014)
26. SaberMand, EidMM: Low power pseudo-random number generator based on lemniscate chaotic map. Int. J. Electr. Comput. Eng. 11(1) (2021)
27. Teng, L., Wang, X., Yang, F., Xian, Y.: Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion. Nonlinear Dyn. 105(2), 1859–1876 (2021)
28. Yang, F., Mou, J., Sun, K., Cao, Y., Jin, J.: Color image compression-encryption algorithm based on fractional-order memristor chaotic circuit. IEEE Access 7, 58751–58763 (2019)
29. Tanveer, M., Shah, T., Rehman, A., Ali, A., Siddiqui, G.F., Saba, T., Tariq, U.: Multi-images encryption scheme based on 3D chaotic map and substitution box. IEEE Access 9, 73924–73937 (2021)
30. Hasanzadeh, E., Yaghoobi, M.: A novel color image encryption algorithm based on substitution box and hyper-chaotic system with fractal keys. Multimed. Tools Appl. 79, 1–19 (2019)
31. Khan, M., Masood, F.: A novel chaotic image encryption technique based on multiple discrete dynamical maps. Multimed. Tools Appl. 78(18), 26203–26222 (2019)

32. Chen, Z., Ye, G.: An asymmetric image encryption scheme based on hash SHA-3, RSA and compressive sensing. Optik 267, 169676 (2022)
33. Teng, L., Wang, X., Yang, F., Xian, Y.: Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion. Nonlinear Dyn. 105(2), 1859–1876 (2021)
34. Hosny, K.M., Kamal, S.T., Darwish, M.M.: Novel encryption for color images using fractional-order hyperchaotic system. J. Ambient Intell. Humaniz. Comput. 13(2), 973–988 (2022)
35. Li, S., Ding, W., Yin, B., Zhang, T., Ma, Y.: A novel delay linear coupling logistics map model for color image encryption. Entropy 20(6), 463 (2018)
36. Wang, M., Liu, H., Zhao, M.: Construction of a nondegeneracy 3D chaotic map and application to image encryption with keyed S-box. Multimed. Tools Appl. 82(22), 34541–34563 (2023)
37. İhsan, Ayşegül, and Nurettin Doğan.: An innovative image encryption algorithm enhanced with the Pan-Tompkins Algorithm for optimal security. Multimedia Tools and Applications 83(35): 82589-82619 (2024)
38. Younas, I., Khan, M.: A new efficient digital image encryption based on inverse left almost semi group and Lorenz chaotic system. Entropy 20(12), 913 (2018)
39. Luengo, E.A., Alaña, B.O., Garc´ıa, L.J., Hernández-Castro, J.: Further analysis of the statistical independence of the NIST SP 800-22 randomness tests. Appl. Math. Comput. 459, 128222 (2023)
40. Richman, J.S., Moorman, J.R.: Physiological time-series analysis using approximate entropy and sample entropy. Am. J. Physiol.-Heart Circ. Physiol. 278(6), H2039–H2049 (2000)
41. Lai, Q., Hu, G., Erkan, U., Toktas, A.: A novel pixel-split image encryption scheme based on 2D Salomon map. Expert Syst. Appl. 213, 118845 (2023)
42. Cao, W., Mao, Y., Zhou, Y.: Designing a 2D infinite collapse map for image encryption. Signal Process. 171, Article 107457 (2020)
43. Gao, X., Yu, J., Banerjee, S., Yan, H., Mou, J.: A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion. Sci. Rep. 11(1), 1–21 (2021)
44. Sun, J.: 2D-SCMCI hyperchaotic map for image encryption algorithm. IEEE Access 9, 59313–59327 (2021)
45. Nan, S., Feng, X., Wu, Y., Zhang, H.: Remote sensing image compression and encryption based on block compressive sensing and 2D-LCCCM. Nonlinear Dyn. 108(3), 2705–2729 (2022)
46. Alexan, W., ElBeltagy, M., Aboshousha, A.: RGB image encryption through cellular automata, S-Box and the Lorenz system. Symmetry 14(3), 443 (2022)
47. Wu, Y., Noonan, J.P., Agaian, S., et al.: NPCR and UACI randomness tests for image encryption. Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun. 1(2), 31–38 (2011)
48. Murillo-Escobar, M.A., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R.M., Acosta Del Campo, O.R.: A RGB image encryption algorithm based on total plain image characteristics and chaos. Signal Process. 109, 119–131 (2015)
49. Liu, P., Wang, X., Su, Y.: Image encryption via complementary embedding algorithm and new spatiotemporal chaotic system. IEEE Trans. Circuits Syst. Video Technol. 33(5), 2506–2519 (2023)
50. Cun, Q., Tong, X., Wang, Z., Zhang, M.: A new chaotic image encryption algorithm based on dynamic DNA coding and RNA computing. Vis. Comput. 39(12), 6589–6608 (2023)
51. Qian, X., Yang, Q., Li, Q., Liu, Q., Wu, Y., Wang, W.: A novel color image encryption algorithm based on three-dimensional chaotic maps and reconstruction techniques. IEEE Access 9, 61334–61345 (2021)
52. Wu, Y., Zhang, L., Berretti, S., & Wan, S.: Medical image encryption by content-aware DNA computing for secure healthcare. IEEE Transactions on Industrial Informatics, 19(2), 2089–2098 (2022)
53. Wei, D., Jiang, M., & Deng, Y.: A secure image encryption algorithm based on hyper-chaotic and bit-level permutation. Expert Systems with Applications, 213, 119074 (2023)
54. Rahul, B., Kuppusamy, K., & Senthilrajan, A.: Dynamic DNA cryptography-based image encryption scheme using multiple chaotic maps and SHA-256 hash function. Optik, 289, 17125 (2023)
55. Yousif, S. F., Abboud, A. J., & Alhumaima, R. S.: A new image encryption based on bit replacing, chaos and DNA coding techniques. Multimedia Tools and Applications, 81(19), 27453–27493 (2022)
56. Zhou, W., Wang, X., Wang, M., & Li, D.: A new combination chaotic system and its application in a new bit-level image encryption scheme. Optics and Lasers in Engineering, 149, 106782 (2022)
57. Niu, Y., Zhou, Z., & Zhang, X.: An image encryption approach based on chaotic maps and genetic operations. Multimedia Tools and Applications, 79(35), 25613–25633 (2020)
58. Gao, M., Li, J., Di, X., Li, X.,& Zhang, M.: A blind signature scheme for IoV based on 2D-SCML image encryption and lattice cipher. Expert Systems with Applications, 246, 123215 (2024)