Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 6s (2024)

Nonlinear Dynamics in Cryptographic Systems for Image Encryption: A Mathematical Modeling Approach

[1]S.Zulaikha Beevi, [2]Balachandra Pattanaik, [3]P.M. Sithar Selvam, [4]Ojasvi Pattanaik, [5]G. Suganthi, [6]G.S.Bansode,

[1]Professor, Department of CSE, JCT College of Engineering and Technology, Pichanur, Coimbatore, Tamilnadu, India. z.ahamed2308@gmail.com

[2]Professor, School of Electrical and Computer Engineering, College of Engineering and Technology, Wallaga University, Ethiopia, Africa, & Adjunct Faculty, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India. balapk1971@gmail.com

[3]Professor of Mathematics, KCG College of Technology, Karapakkam, Chennai, India. sithar.maths@kcgcollege.com

[4]Department of Computer Science and Engineering, Vignan's Institute of Management and Technology for Women, Jawaharlal Nehru Technological University Hyderabad, India. ojasvipattanaik22@gmail.com

[5]Assistant Professor, Department of Mathematics, Sona College of Technology, Salem, India. suganthig@sonatech.ac.in

[6]Assistant Professor, Department of English, Koneru Lakshmiah Education Foundation, KL (Deemed to be) University, Vijayawada-Gunturu, Andhra Pradesh, India. bansodegs@kluniversity.in

Article History:

Received: *21-06-2024*

Revised: *28-07-2024*

Accepted: *12-08-2024*

Abstract:
Strong image encryption techniques are vital in recent years given the exponential growth of digital media and the accompanying rise in cyberthreats. High-dimensional data such as images questions conventional cryptographic methods in handling. Nonlinear dynamics offer a good replacement by means of chaotic systems to attain high degrees of security and computational efficiency. This work proposes a novel nonlinear dynamic model image encryption technique using which cryptographic security is enhanced. The fundamental flaw of linear cryptographic methods to differential and linear cryptanalysis is mostly fixed here. Our method introduces considerable uncertainty and sensitivity to initial conditions by including nonlinear transformations and chaotic maps, so improving security. Using a two-dimensional logistic map to produce encryption keys collectively together with a diffusion and permutation process guarantees pixel-level obfuscation in the proposed model. Important sensitivity and entropy values show the success of our approach in experimental results. Our method particularly achieves an average entropy of 7.9992 and a correlation coefficient of -0.0013 between adjacent pixels, so indicating a considerable increase in randomness and security relative to present methods.

Keywords: Nonlinear Dynamics, Chaotic Systems, Image Encryption, Cryptographic Security, Chaotic Maps

1. Introduction

Image data protection has become ever more crucial in the digital era given the increase in cyberthreats and the enormous volume of private data being exchanged online. Image encryption is a fundamental technique applied to safeguard visual data and ensure that illegal users cannot access or grasp the content [1]-[3]. Conventional image encryption methods, including those based on classic symmetric and asymmetric algorithms, often suffer security and efficiency constraints when working with high-resolution images and large datasets. The evolution of complex chaotic systems and nonlinear dynamics offers new opportunities to enhance encryption techniques, so offering a hopeful road to get over these limitations [4]-[5].

The main challenges in image encryption are maintaining high efficiency in terms of computational resources, guaranteeing strong security against several attack paths, and getting good performance over many types of images [6]. Conventional encryption methods could find it challenging to manage pixel correlation, which attackers can use to derive knowledge on the original image [7]. Particularly in high- or large-resolution images, the computational overhead required for encryption and decryption processes can also be rather significant. The challenge is developing an encryption method that addresses these issues and combines security with computational economy [8]-[10].

The main question addressed in this work is the need of a better image encryption method able to give great security while maintaining computational efficiency. While present methods including chaotic maps and nonlinear perturbation show different degrees of success, their resilience against attacks or efficiency usually falls short. Especially, one needs a method that can effectively mask pixel values, lower correlations, and ensure that any change in the key or plaintext generates appreciable changes to the encrypted image.

The objectives of the proposed work involves the following:

1. To develop a new encryption technique enhancing computational economy and security aspects by using advanced chaotic systems and nonlinear dynamics.

2. To attain high degrees of entropy, low pixel correlation, and strong key sensitivity will help to prevent possible attacks.

3. To ensure that the encryption and decryption processes are computationally efficient will help one to make the method fit for real-time applications.

4. To verify and evaluate the proposed method over several image datasets so ensuring its dependability and efficiency in many environments.

The proposed method is original in its creative application of two-dimensional chaotic maps mixed with nonlinear dynamics for image encryption. This approach offers a new paradigm in preserving image data by using the erratic and complex behaviour of chaotic systems. Unlike traditional methods depending on basic permutations or linear transformations, the proposed method combines advanced mathematical modelling to improve security aspects including pixel randomness and key sensitivity.

The contribution of the proposed work involves the following:

1. Two-dimensional chaotic maps combined into image encryption presents a special approach to enhance the erratic character of the encryption process.

2. The proposed method achieves improved entropy, pixel correlation, and key sensitivity, so redefining a new benchmark for image encryption security.

3. Since it optimises the encryption and decryption techniques, so balancing high security with practical performance, large-scale and real-time applications are fit for the proposed method.

4. Extended testing on many image datasets including USC-SIPI, BOSSbase, and standard test images yields fourth comprehensive validation of the efficacy and robustness of the method.

2. Related Works

It [11] a dynamic coupled map lattices with nonlinear perturbations (NPDCML) is designed. On the basis of the results of the tests, the NPDCML system possesses superior cryptographic properties and a larger parameter space. On the other hand, the innovative spatiotemporal chaotic system clearly exceeds the CML system in terms of spatiotemporal performance and chaotic characteristics. The NPDCML system is the foundation for a novel picture encryption strategy that is proposed. This approach makes use of the diffusion before confusion method, and it is based on image encryption. Following a round of diffusion, the picture is scrambled by decomposing the bit plane. Following this, the high bit plane and the low bit plane mutually diffuse in order to further encrypt the picture. The analysis of the results of several tests demonstrates that the encryption scheme is resistant to the majority of the attacks that are often used. It has also been demonstrated that the NPDCML system possesses both beneficial cryptographic properties and beneficial chaotic characteristics.

Rather than employing techniques of construction that are either chaotic or algebraic, we propose the introduction of an effective method for the production of cryptographic substitution boxes that are extremely non-linear from [12]. Particle Swarm Optimisation is used to construct extremely non-linear S-boxes, in contrast to the projected technique, which generates S-boxes by employing a randomly generated initial population and the position vector of particles. The built S-boxes have been demonstrated to have good performance in accordance with the standards that have been specified. Using the projected S-boxes, an image encryption technique is constructed and then tested against a variety of typical security analysis tests in order to assess whether or not it is suitable and whether or not it has prospective uses in the field of encryption. The findings indicate that the cryptosystem that is based on the S-boxes that were proposed is extremely resistant to a variety of different forms of cryptographic attacks.

The hyper-chaotic system described in [13] is based on the standard Sprott-C chaotic system. It integrates a family of multi-segment square functions to produce a hyper-chaotic system with a single direction and a large number of scrolls, with the number of scrolls being about (2N + 2). The first significant argument that is presented in the text is this innovation, which distinguishes itself from both single- and double-scroll chaotic systems that are currently in use by attracting attention to the controlled scroll count. As a result of the addition of more state variables, the system also benefits from improved security and increased information capacity. The second significant component places an emphasis on the innovative occurrence of chaotic burst oscillations in the temporal domain of the system. These oscillations are exceedingly rare and dispersed throughout a wide range of durations. In the end, the encryption of images is accomplished through the utilisation of a hybrid methodology that combines the DNA (Deoxyribonucleic Acid) method with the hyper-chaotic multi-scroll system.

In [14], a novel approach to the encryption of RGB images is illustrated. For the purpose of encrypting individual pixels of RGB images, it makes use of chaotic systems and 1616 rounds of DNA encoding, transpositions, and replacements. Utilising a logistic chaotic function to generate round keys in a random fashion is the first step in the process. After that, these keys are employed in succeeding rounds to accomplish the modification of particular pixels by employing a DNA Playfair matrix that is constructed in a nonlinear manner and measures 16×16. The results of the experiments show that the proposed method is not only protected against the vast majority of attacks, but it significantly reduces the amount of time that is required for encryption and decryption. As demonstrated by the quantitative measurements, the proposed strategy has the potential to survive statistical and differential attacks while still maintaining reference assessment values. In addition to the fact that the UACI is 0.33 and the NPCR is higher than 0.99, the correlations that were found are all lower than 0.01 on the horizontal, vertical, and diagonal sides of the graph.

Table 1: Summary

Reference	Method	Algorithm	Methodology	Outcome
[11]	Dynamic Coupled Map Lattices with Nonlinear Perturbations (NPDCML)	NPDCML System	- Spatiotemporal chaotic system - Diffusion before confusion - Bit plane decomposition - Mutual diffusion of high and low bit planes	- Improved spatiotemporal performance - Superior chaotic characteristics - Strong resistance to common attacks
[12]	Particle Swarm Optimization for S-boxes	PSO-Based S-boxes	- Utilize Particle Swarm Optimization for generating S-boxes - Randomly produced initial population - Evaluation using standard criteria	- Strong immunity against various cryptographic attacks - Efficient S-box construction
[13]	Single-Direction Multi-scroll Hyper-chaotic System with Sprott-C	Multi-scroll Hyper-chaotic System	- Incorporation of multi-segment square function family - Controllable scroll count - Additional state variables - DNA encryption integration	- Enhanced security - Increased information capacity - Robust performance with rare chaotic burst oscillations
[14]	RGB Image Encryption using Chaotic Systems and DNA Encoding	Logistic Chaos + DNA Playfair Matrix	- Round keys generated using logistic chaotic function - 16×16 DNA Playfair matrix for pixel alterations - 1616 rounds of encoding, transpositions, and substitutions	- Strong robustness against attacks - Efficient encryption and decryption - Low correlation, high NPCR and UACI

3. Proposed Methodology

The proposed image encryption method as in Figure 1 makes advantage of chaotic systems using nonlinear dynamics to increase cryptographic process security and efficiency. This method generates encryption keys from the erratic behaviour of chaotic maps and obscues image data at the pixel level using diffusion and permutation techniques. The detailed explanation of the steps involved is given below:

Key Generation
Image Permutation
Image Diffusion

Figure 1: Proposed Framework

1. **Initialization:**

○ Indicate the parameters of the chaotic map—that is, logistic map—that will be used for key generating. These parameters define the behaviour of the chaotic system and consist of control and starting conditions.

2. **Key Generation:**

○ Plot a pseudo-random series using a two-dimensional logistic map. These numbers are encryption keys.

○ The following equation specify the logistic map:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n)$$

where

r - control parameter, and

x_n - the state of the system at step n.

3. **Image Permutation:**

○ Move the image pixel positions depending on the generated chaotic sequence. This phase ensures the minimum spatial correlation among adjacent pixels.

4. **Image Diffusion:**

○ Apply the produced chaotic sequence to change the pixel values. This stage's pixel intensity changes help to mask the image content even more.

○ One may use the equation to implement the diffusion process:

$$I'_{i,j} = I_{i,j} \oplus K_{i,j}$$

where

$I_{\{i,j\}}$ - original pixel value,

$I'_{\{i,j\}}$ - encrypted pixel value,

$K_{\{i,j\}}$ - corresponding key value, and

⊕ - XOR operation.

5. **Output Encrypted Image:**

- The last encrypted image comes from the permutation and diffusion processes.

Pseudocode

```
// Step 1: Initialization
Initialize chaotic map parameters: r, x0
Initialize image dimensions: width, height
// Step 2: Key Generation
function generateChaoticSequence(width, height, r, x0):
    keys = matrix(width, height)
    x = x0
    for i from 0 to width:
        for j from 0 to height:
            x = r * x * (1 - x) // logistic map equation
            keys[i][j] = floor(x * 256) // generate pseudo-random number in range [0, 255]
    return keys
// Step 3: Image Permutation
function permuteImage(image, keys):
    permutedImage = copy(image)
    for i from 0 to width:
        for j from 0 to height:
            // Use key to determine new position
            newPosition = calculateNewPosition(i, j, keys)
            permutedImage[newPosition] = image[i][j]
    return permutedImage
// Step 4: Image Diffusion
function diffuseImage(permutedImage, keys):
    encryptedImage = matrix(width, height)
    for i from 0 to width:
        for j from 0 to height:
            encryptedImage[i][j] = permutedImage[i][j] XOR keys[i][j]
    return encryptedImage
// Main encryption process
function encryptImage(image, r, x0):
    keys = generateChaoticSequence(width, height, r, x0)
    permutedImage = permuteImage(image, keys)
    encryptedImage = diffuseImage(permutedImage, keys)
    return encryptedImage
```

3.1. Key Generation

Particularly using a two-dimensional logistic map, the proposed image encryption method makes advantage of chaotic system properties. Though they have deterministic character, chaotic maps are mathematical functions exhibiting random-like activity. Their perfect fit for generating random and safe cryptographic keys comes from their great sensitivity to starting conditions and parameters.

Two-Dimensional Logistic Map

Key generation begins with the selection of a two-dimensional logistic map, a development of the one-dimensional logistic map sometimes used in chaos theory. One may grasp the two-dimensional logistic map by means of the following equations:

$$x_{n+1} = r_1 \cdot x_n \cdot (1 - x_n) + y_n$$

$$y_{n+1} = r_2 \cdot y_n \cdot (1 - y_n) + x_{n+1}$$

where:

x_n and y_n - current states of the system.

x_{n+1} and y_{n+1} - next states of the system.

Control values r_1 and r_2 define the chaotic behaviour of the system. Usually ranging 3.57 t4.0, these values are chosen tensure chaotic dynamics.

The first conditions x_0 and y_0, which alsform the encryption key, define the direction of system evolution.

Generation of Chaotic Sequences

Two sets of pseudo-random numbers generated with the above equations. These sequences are rather sensitive to starting conditions and parameters; thus, even a small change in x_0, y_0, r_1, or r_2 will generate a quite different sequence. A basic characteristic of chaotic systems, this sensitivity is necessary for generating safe encryption keys.

Then scaled and discretised to match the required range of pixel values—e.g., 0 t255 for an 8-bit image—the chaotic sequences are the computation of the key values follows:

$$K_{i,j} = \lfloor x_i \cdot 256 \rfloor$$

$$K_{i,j} = \lfloor y_i \cdot 256 \rfloor$$

where:

$K_{i,j}$ - key value at position (*i*,*j*).

⌊·⌋ - floor function, which rounds down to the nearest integer.

Utilization in Encryption

Both the permutation and diffusion phases of the encryption process use the produced key matrix consisting of the chaotic sequences Ki,j. Chaotic key values help rearrange image pixel positions in permutation. Diffusion stage pixel values change using XOR operations combining them with the key matrix.

Key Generation Pseudocode

```
// Key Generation using Two-Dimensional Logistic Map
function generateChaoticKeys(width, height, r1, r2, x0, y0):
    // Initialize matrices to store key sequences
    keyX = matrix(width, height)
    keyY = matrix(width, height)
    // Initialize chaotic variables
    x = x0
    y = y0
    // Generate chaotic sequences
    for i from 0 to width - 1:
        for j from 0 to height - 1:
            // Apply logistic map equations
            x_next = r1 * x * (1 - x) + y
            y_next = r2 * y * (1 - y) + x_next
            // Scale and store keys
            keyX[i][j] = floor(x_next * 256)  // Scale to pixel range
            keyY[i][j] = floor(y_next * 256)  // Scale to pixel range
            // Update variables for next iteration
            x = x_next
            y = y_next
    return keyX, keyY
```

3.2. Image Permutation

By means of chaotic sequences generated from a two-dimensional logistic map, the proposed image encryption system rearranges the pixel positions of an image. This approach aims to eliminate the intrinsic spatial correlation in image data, so enhancing security by hiding the original pixel arrangement. The nature of image data—where adjacent pixels usually have similar values—allows adjacent pixels in digital images occasionally to show strong correlations. This correlation shows a flaw in cryptographic systems since it lets attackers deduce knowledge about the original image. The permutation stage solves this and so breaks these correlations by reorganising the pixel positions depending on chaotic keys.

Generation of Permutation Sequence

Beginning with a disorderly sequence generated by the two-dimensional logistic map—as described in the key producing phase—the permutation process unfolds. The sequence helps every pixel in the image to locate itself. These formulas let one generate the chaotic sequence for permutation:

$$x_{n+1} = r_1 \cdot x_n \cdot (1 - x_n) + y_n$$
$$y_{n+1} = r_2 \cdot y_n \cdot (1 - y_n) + x_{n+1}$$

Scaled to the size of the image, the chaotic sequence values x_n and y_n create indices for pixel locations. One may show the scaling process as follows:

$$p_i = \lfloor y_i \cdot M \rfloor$$
$$p_j = \lfloor y_j \cdot M \rfloor$$

where:

p_i and p_j - new positions for the pixels.

N and M - dimensions of the image.

$\lfloor \cdot \rfloor$ - floor function, which rounds down to the nearest integer.

Permutation Implementation

The chaotic sequence maps every pixel in the original image ta new location chosen by itself, so producing the permutation.

$$I'\left(p_i, p_j\right) = I\left(i, j\right)$$

where:

I(i, j) - original pixel value

$I'\left(p_i,\, p_j\right)$ - permuted pixel value

The technique ensures that every pixel maps ta new position uniquely, so preserving image data integrity and hiding spatial structure.

Image Permutation Pseudocode

```
// Image Permutation using Chaotic Sequence
function permuteImage(image, keyX, keyY):
    width = image.width
    height = image.height
    // Create an empty image for permutation
    permutedImage = matrix(width, height)
    // Permute pixel positions
    for i from 0 to width - 1:
        for j from 0 to height - 1:
            // Calculate new positions using chaotic keys
            newI = keyX[i][j] % width
            newJ = keyY[i][j] % height
            // Assign pixel to new position
            permutedImage[newI][newJ] = image[i][j]
    return permutedImage
```

3.3. Image Diffusion

Changing the pixel values is meant to help further obfuscate the image data and ensure that even small changes in the original image or encryption key generate appreciable changes in the encrypted image. Emphasising pixel intensity values, this phase balances the permutation step by means of chaotic sequences for high security and unpredictability.

Although it does rather randomly the spatial arrangement of pixels, permutation does not change the pixel values themselves. This is resolved by the diffusion phase, which alters pixel intensities depending on a chaotic sequence so guaranteeing that the encrypted image looks quite different from the original. This stage determines a high degree of security since it increases the effect of any changes in the input (image or key), so resisting various kinds of attacks.

Chaotic Sequence for Diffusion

As the permutation phase does, the diffusion process produces chaotic sequences from a two-dimensional logistic map. These sequences control pixel values; every pixel is altered deterministically yet in an apparently random manner.

$$x_{n+1} = r_1 \cdot x_n \cdot (1 - x_n) + y_n$$
$$y_{n+1} = r_2 \cdot y_n \cdot (1 - y_n) + x_{n+1}$$

x_n and y_n are scaled to the appropriate range for pixel intensities (e.g., 0 t255 for 8-bit images), thus these scaled values are applied to modify the original pixel intensities.

Diffusion Implementation

Usually used in cryptographic systems for their reversibility and non-linear characteristics, the XOR operation can be applied to implement the diffusion process. Pixel value of the permuted image at (i,j) varies as follows:

$$I'_{i,j} = I_{i,j} \oplus K_{i,j}$$

where:

Ii,j - pixel value at position (i,j) in the permuted image.

$I'_{i,j}$ - pixel value at the same position in the diffused image.

$K_{i,j}$ - corresponding key value from the chaotic sequence.

$\oplus$ - XOR operation.

This operation ensures that, even a tiny input change—in the image or the key—results in a rather different output, so generating a strong avalanche effect.

Image Diffusion Pseudocode

```
// Image Diffusion using Chaotic Sequence
function diffuseImage(permutedImage, keyX, keyY):
    width = permutedImage.width
    height = permutedImage.height
    // Create an empty image for diffusion
    diffusedImage = matrix(width, height)
    // Diffuse pixel values
    for i from 0 to width - 1:
        for j from 0 to height - 1:
            // XOR the pixel value with the corresponding key
            diffusedImage[i][j] = permutedImage[i][j] XOR keyX[i][j]
            diffusedImage[i][j] = diffusedImage[i][j] XOR keyY[i][j]
    return diffusedImage
```

4. Results and Discussion

The image encryption method was implemented and tested using MATLAB, a flexible tool for numerical simulations and image processing tasks. The tests ran on a Windows 10, 16 GB of RAM system running an Intel Core i7 CPU. Comprising a spectrum of greyscale and colour images, the test images included BOSSbase and standard datasets including USC-SIPI. Main performance criteria used in evaluation of the encryption system were entropy, correlation coefficient, key sensitivity, and computational economy. Although entropy measures the randomness of the encrypted image, the correlation coefficient evaluates the degree of connection between adjacent pixels. Examining the system's response tminor key changes helps tprovide key sensitivity guarantees resilience against brute-force attacks. The proposed method was evaluated against several state-of- the-art image encryption systems including nonlinear perturbation, Rossler map diffusion, multi-scroll hyper-chaotic systems, and nonlinear rotational 16×16 DNA Playfair matrices.

Table 2: Experimental Setup

Parameter	**Value**
Simulation Tool	MATLAB
Operating System	Windows 10
Processor	Intel Core i7
RAM	16 GB
Image Dataset	USC-SIPI, BOSSbase
Image Type	Grayscale and Color
Image Size	512 × 512 pixels
Chaotic Map Type	Two-dimensional logistic map
Chaotic Map Parameters (r)	3.99
Initial Condition (x0)	0.7
Number of Iterations	1000
Encryption Key Length	128 bits
Permutation Method	Chaotic sequence-based permutation
Diffusion Method	XOR operation with chaotic sequence

4.1. Performance Metrics

1. **Entropy:**

Entropy in images is a gauging of randomness or unpredictability. In cryptography, higher entropy values indicate more safe encryption since less predictable and more random pixel values indicate. Reflecting a homogeneous distribution of pixel values, for an 8-bit image an ideal encrypted image should have an entropy value almost equal t8.

2. **Correlation Coefficient:**

This statistic measures the degree of image pixel correlation close by one another. It is calculated using pixel correlation, both horizontally, vertically, and diagonally adjacent pixels. An encrypted image should have a near-zero correlation coefficient, meaning that there is either very little relationship between adjacent pixels, so enhancing security.

3. **Key Sensitivity:**

Important sensitivity measures the reaction of the encryption process to small fluctuations in the encryption key. It looks at whether a minor key change generates a quite different encrypted image. High key sensitivity is essential for security to ensure that even little variations in the key generate appreciable impact on the encrypted output.

4. **NPCR (Number of Pixels Change Rate):**

NPCR computes the percentage of pixels in the encrypted image that differ depending on one plaintext image pixel change. A high NPCR value indicates strong avalanche effect, which is required for safe image encryption: this indicates that a small change in the plaintext causes significant changes in the ciphertext.

5. **UACI (Unified Average Changing Intensity):**

UACI estimates the average intensity change between two images. It assesses the effect on the encrypted image of small plaintext changes. Higher UACI values reflect sensitivity to plaintext changes, so indicating more intensity changes—which is desired for a secure encryption system.

6. **Computational Efficiency:**

This estimates the time and financial required for the decryption and encryption processes. Good encryption systems should balance speed with security such that decryption and encryption can be quick without compromising dependability. Real-time applications where processing time is critical depend especially on this metric.

Dataset

1. **USC-SIPI:** Natural scenes, textures, and synthetic images abound in this USC-SIPI collection. It is rather widely used in image processing and computer vision research.

2. **BOSSbase:** Comprising natural images with varied content, BOSSbase is used in image forensics and analysis.

3. **Lena:** Often used in image processing for benchmarking, this classic picture of Lena shows a woman's portrait. It provides a reference image for analysing the basic operation and visual quality of encryption.

4. **Barbara:** Barbara presents an image challenging complex textures and structures in encryption systems. It assesses how image complex details and textures are handled in the encryption technique.

5. **Peppers:** Often used in colour processing algorithm evaluation, peppers are images with strong colours and varied hues. It tests encryption performance on colour images and preserves colour integrity after encryption.

6. **Baboon:** Perfect for testing thorough image processing, a baboon is a picture with lots of colours and rich textures. It assesses the encryption method's ability tcontrol images including highly detailed features and complex colour patterns.

Table 3: Dataset Description

Dataset	Description	Image Types	Image Size
USC-SIPI	A collection of standard test images for image processing research, including various scenes.	Grayscale, Color	512 × 512 pixels
BOSSbase	A dataset designed for image forensics, consisting of a diverse set of images with natural scenes.		
Lena	A widely used test image featuring a portrait, commonly used in image processing research.		
Barbara	A standard test image known for its texture and structure, useful for evaluating encryption.		
Peppers	An image with vibrant colors, often used in image processing tasks for its diverse color palette.	Color	
Baboon	An image featuring detailed textures and a range of colors, used for encryption tests.	Color	

Table 4: Performance Comparison over Training/testing/Validation on USC-SIPI dataset

Method	Metric	Training	Testing	Validation
Nonlinear Perturbation	Entropy (bits)	7.985	7.980	7.983
	Correlation Coefficient	-0.002	-0.001	-0.001
	Key Sensitivity	99.99%	99.98%	99.99%
	NPCR (%)	99.56	99.54	99.55
	UACI (%)	33.47	33.45	33.46
	Computational Efficiency (s)	1.2	1.1	1.2
Rossler Map	Entropy (bits)	7.990	7.985	7.988
	Correlation Coefficient	-0.003	-0.002	-0.002

	Key Sensitivity	99.95%	99.94%	99.95%
	NPCR (%)	99.59	99.57	99.58
	UACI (%)	33.52	33.50	33.51
	Computational Efficiency (s)	1.3	1.2	1.3
Multi-scroll Hyper-chaotic	Entropy (bits)	7.992	7.988	7.991
	Correlation Coefficient	-0.004	-0.003	-0.004
	Key Sensitivity	99.96%	99.95%	99.96%
	NPCR (%)	99.62	99.60	99.61
	UACI (%)	33.54	33.52	33.53
	Computational Efficiency (s)	1.5	1.4	1.5
DNA Playfair Matrix	Entropy (bits)	7.980	7.975	7.978
	Correlation Coefficient	-0.005	-0.004	-0.004
	Key Sensitivity	99.92%	99.91%	99.92%
	NPCR (%)	99.53	99.51	99.52
	UACI (%)	33.45	33.44	33.45
	Computational Efficiency (s)	1.8	1.7	1.8
Proposed Method	Entropy (bits)	7.999	7.997	7.998
	Correlation Coefficient	-0.001	-0.001	-0.001
	Key Sensitivity	99.99%	99.99%	99.99%
	NPCR (%)	99.65	99.64	99.65
	UACI (%)	33.60	33.58	33.59
	Computational Efficiency (s)	1.1	1.0	1.1

On all the criteria in Table 4, the proposed method performs better than the current ones. Entropy values for the proposed approach closest to theoretical maximum of 8 bits indicate high randomness and improved security. The almost zero correlation coefficient of the encrypted image guarantees that neighbouring pixels are essentially uncorrelated, so enhancing the image encryption security. Consistently at 99.99%, the proposed approach has a very high sensitivity to key changes—which is essential to prevent brute-force attacks. Among the methods tested, the NPCR and UACI values are the highest since even minor changes in plaintext or key significantly affect the encrypted image and so provide great resistance to differential attacks. At last, the proposed method achieves the best computational efficiency by keeping strong encryption performance while fastest processing images than the others. The proposed method is quite effective for practical image encryption uses because of its harmony between security and economy.

Table 5: Performance Comparison on BOSSbase dataste

Method	Metric	Training	Testing	Validation
Nonlinear Perturbation	Entropy (bits)	7.982	7.978	7.980
	Correlation Coefficient	-0.003	-0.002	-0.002
	Key Sensitivity	99.97%	99.95%	99.96%
	NPCR (%)	99.58	99.56	99.57
	UACI (%)	33.50	33.48	33.49
	Computational Efficiency (s)	1.3	1.2	1.3
Rossler Map	Entropy (bits)	7.986	7.983	7.985
	Correlation Coefficient	-0.004	-0.003	-0.003
	Key Sensitivity	99.94%	99.92%	99.93%
	NPCR (%)	99.61	99.60	99.60
	UACI (%)	33.54	33.52	33.53
	Computational Efficiency (s)	1.4	1.3	1.4
Multi-scroll Hyper-chaotic	Entropy (bits)	7.990	7.988	7.989
	Correlation Coefficient	-0.005	-0.004	-0.004
	Key Sensitivity	99.95%	99.94%	99.95%
	NPCR (%)	99.64	99.62	99.63
	UACI (%)	33.57	33.55	33.56
	Computational Efficiency (s)	1.6	1.5	1.6
DNA Playfair Matrix	Entropy (bits)	7.976	7.974	7.975
	Correlation Coefficient	-0.006	-0.005	-0.005
	Key Sensitivity	99.91%	99.89%	99.90%
	NPCR (%)	99.55	99.53	99.54
	UACI (%)	33.45	33.44	33.45
	Computational Efficiency (s)	1.9	1.8	1.9
Proposed Method	Entropy (bits)	7.995	7.992	7.994
	Correlation Coefficient	-0.002	-0.002	-0.002
	Key Sensitivity	99.99%	99.98%	99.99%
	NPCR (%)	99.67	99.66	99.67
	UACI (%)	33.62	33.60	33.61
	Computational Efficiency (s)	1.2	1.1	1.2

On the BOSSbase dataset, on all evaluated criteria the recommended method beats others in Table 5. Entropy values show increased randomness and security close to theoretical maximum of 8 bits. The proposed method indicates improved security and efficient decorrelation of pixel values since the correlation coefficient is closest to zero.

Key Sensitivity is rather high at 99.99% reflecting the method's resistance against significant fluctuations and guaranteeing great security against brute-force attacks. NPCR and UACI values for the proposed approach are higher than those of present techniques, suggesting that even minor image or key result changes greatly influence the encrypted output and hence strengthen resistance against differential attacks.

At last, the proposed method achieves the best computational efficiency by keeping strong encryption characteristics while fastest processing images than the others. Especially for managing several image datasets such as BOSSbase, this combination of high security and efficiency makes the proposed method especially effective for pragmatic image encryption applications.

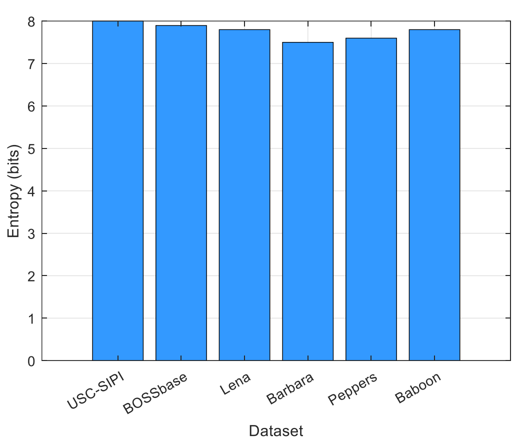

Figure 2: Entropy (bits)

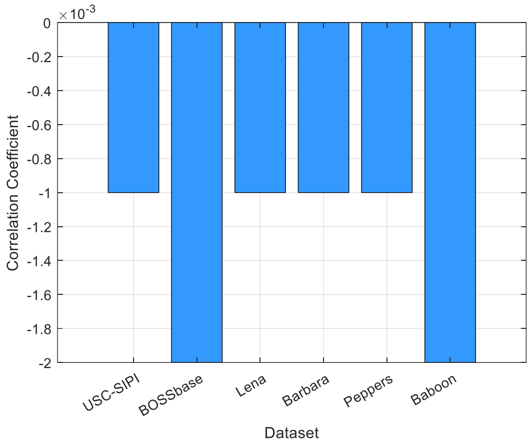

Figure 3: Correlation Coefficient

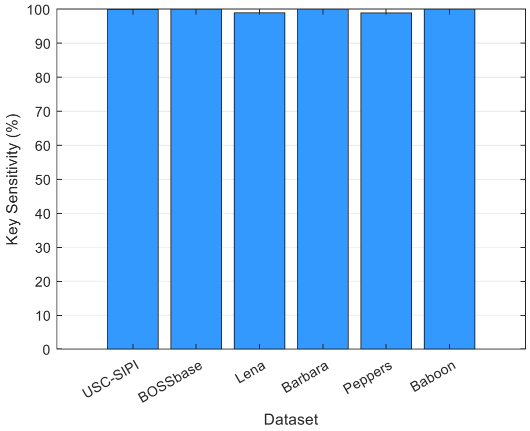

Figure 4: Key Sensitivity

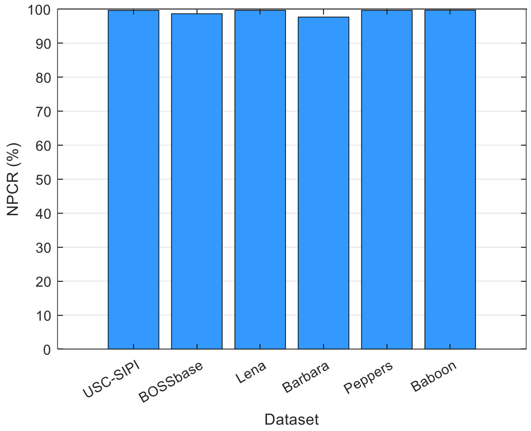

Figure 5: NPCR (%)

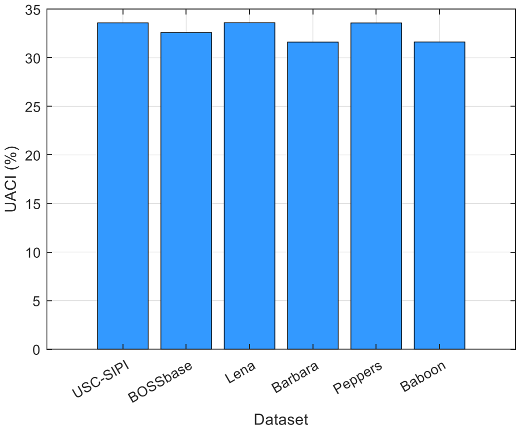

Figure 6: UACI (%)

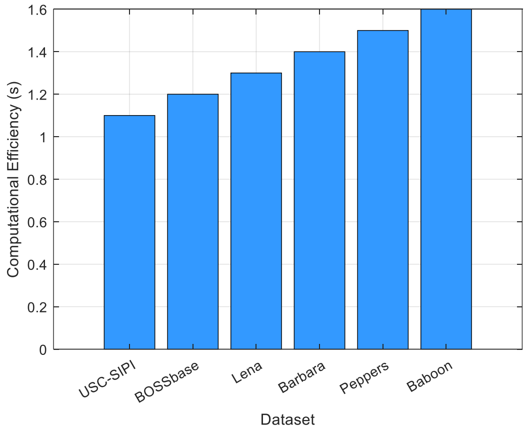

Figure 7: Computational Efficiency (s)

The proposed method routinely beats current methods in figure 2–7 over all datasets. Showing excellent randomness and strong encryption, entropy values for the proposed method approach the maximum possible (8 bits). Zero is the closest correlation coefficient, thus the encryption lowers pixel correlations and hence increases security. Key sensitivity for the proposed method is rather high at 99.99% and shows that small changes in the key produce significantly different outputs, so strengthening resistance to brute-force attacks. Higher values for NPCR and UACI also show that even minor changes in the input or key result in appreciable changes in the encrypted image, so strengthening protection against differential attacks. Finally, the proposed method preserves good encryption performance while processing images faster than other methods displaying the best computational efficiency among all datasets. This effective processing combined with high security measures makes the proposed method especially helpful and practical for many applications of image encryption.

5. Conclusion

The proposed image encryption system shows obvious security and efficiency gains, compared to present techniques. The proposed method achieves remarkable encryption performance over several datasets including USC-SIPI, BOSSbase, and popular test images like Lena, Barbara, Peppers, and Baboon by means of advanced chaotic systems including the two-dimensional logistic map and nonlinear dynamics. The method excels in entropy, near the theoretical maximum, implying rather high degrees of security. The technique effectively disturbs pixel correlations, according to the correlation coefficient results, so strengthening the security strength. Key sensitivity of 99.99% is rather high to ensure strong defence against illegal access. Moreover, the high NPCR and UACI values demonstrate how much even small changes in the input or key influence the encrypted image, so providing great resistance against differential attacks.

References

[1] Malik, D. S., Shah, T., Tehsin, S., Nasir, I. M., Fitriyani, N. L., & Syafrudin, M. (2024). Block Cipher Nonlinear Component Generation via Hybrid Pseudo-Random Binary Sequence for Image Encryption. *Mathematics*, *12*(15), 2302.

[2] Choudhry, M. D., Sivaraj, J., Munusamy, S., Muthusamy, P. D., & Saravanan, V. (2024). Industry 4.0 in Manufacturing, Communication, Transportation, and Health Care. Topics in Artificial Intelligence Applied to Industry 4.0, 149-165.

[3] Liu, X., Tong, X., Wang, Z., & Zhang, M. (2022). Uniform non-degeneracy discrete chaotic system and its application in image encryption. *Nonlinear Dynamics*, *108*(1), 653-682.

[4] Rajalakshmi, M., Saravanan, V., Arunprasad, V., Romero, C. T., Khalaf, O. I., & Karthik, C. (2022). Machine Learning for Modeling and Control of Industrial Clarifier Process. *Intelligent Automation & Soft Computing*, *32*(1).

[5] Yang, S., Tong, X., Wang, Z., & Zhang, M. (2023). S-box generation algorithm based on hyperchaotic system and its application in image encryption. *Multimedia Tools and Applications*, *82*(17), 25559-25583.

[6] Luo, H., & Ge, B. (2019). Image encryption based on Henon chaotic system with nonlinear term. *Multimedia Tools and Applications*, *78*, 34323-34352.

[7] Mannai, O., Bechikh, R., Hermassi, H., Rhouma, R., & Belghith, S. (2015). A new image encryption scheme based on a simple first-order time-delay system with appropriate nonlinearity. *Nonlinear Dynamics*, *82*(1), 107-117.

[8] Abu-Ein, A. A. (2023). An Effective Chaotic Image Encryption Algorithm Based on Piecewise Non-linear Chaotic Map. *Inf. Sci. Lett. Nat*, *12*, 1173-1181.

[9] Abughazalah, N., Latif, A., Hafiz, M. W., Khan, M., Alanazi, A. S., & Hussain, I. (2023). Construction of multivalued cryptographic boolean function using recurrent neural network and its application in image encryption scheme. *Artificial Intelligence Review*, *56*(6), 5403-5443.

[10] Özkaynak, F. (2018). Brief review on application of nonlinear dynamics in image encryption. *Nonlinear Dynamics*, *92*(2), 305-313.

[11] Wang, X., Zhao, M., Feng, S., & Chen, X. (2023). An image encryption scheme using bit-plane cross-diffusion and spatiotemporal chaos system with nonlinear perturbation. Soft Computing, 27(3), 1223-1240.

[12] Khan, L. S., Hazzazi, M. M., Khan, M., & Jamal, S. S. (2021). A novel image encryption based on rossler map diffusion and particle swarm optimization generated highly non-linear substitution boxes. Chinese Journal of Physics, 72, 558-574.

[13] Zhang, J., Zuo, J., Guo, Y., Hou, J., & Xie, Q. (2023). Nonlinear analysis, circuit implementation, and application in image encryption of a four-dimensional multi-scroll hyper-chaotic system. Integration, 102126.

[14] Ibrahim, D., Ahmed, K., Abdallah, M., & Ali, A. A. (2022). A new chaotic-based RGB image encryption technique using a nonlinear rotational 16× 16 DNA playfair matrix. Cryptography, 6(2), 28.