



# A Two-Stage Image Encryption Framework using Ensemble Learning-Driven Key Generation and Spiral Ripple Shuffling

Gaddafi Abdul-Salaam<sup>1</sup>, Issah Zabsonre Alhassan<sup>1,2,\*</sup>, Michael Asante<sup>1</sup>, Yaw Marfo Missah<sup>1</sup>, Farkhana Muchtar<sup>3</sup>, Alimatu Sadia Shirazu<sup>1</sup>

<sup>1</sup> Department of Computer Science, Faculty of Physical and Computational Sciences, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

<sup>2</sup> Department of Medical Imaging, School of Allied Health Sciences, University for Development Studies, Tamale, Ghana

<sup>3</sup> Faculty of Computing, Universiti Teknologi Malaysia, Skudai, 81310 Johor Bahru, Johor, Malaysia

## ARTICLE INFO

## ABSTRACT

### Article history:

Received 21 March 2025

Received in revised form 11 April 2025

Accepted 31 July 2025

Available online 11 August 2025

### Keywords:

Image encryption; ensemble learning; spiral ripple shuffle (SRS); pseudorandom key generation; bitwise XOR

The surge in the use of digital imaging demands ciphers that are both mathematically rigorous and computationally light. However, there are gaps in earlier chaos- or Machine Learning-based schemes such as inefficient key generation, limited computational scalability and vulnerabilities to advanced attacks. Particularly when integrating machine learning with cryptographic operations by producing fully key-dependent row/column permutations and diffusion in a single pass. This study therefore introduces a two-stage framework that marries an ensemble-learning-driven pseudorandom key generator with a spiral-ripple shuffle followed by XOR diffusion to dismantle pixel correlations at linear-time complexity  $O(N)$ . Experiments on six benchmark images confirm the design's statistical resilience with averages of NPCR = 99.57 %, UACI = 34.63 %, entropy = 7.52 bits and SSIM  $\approx$  0.01 between cipher and plain images. Recovery fidelity remains high (PSNR up to 53.97 dB), while the heaviest image encrypts in 0.94s and lighter images in  $\approx$  0.03s on standard desktop hardware. These figures indicate near-ideal diffusion, uniform histogram distribution and negligible perceptual leakage, outperforming recent chaos-IoT ciphers in runtime without sacrificing security metrics. Therefore, the proposed system achieves real-time throughput for megapixel frames, positioning it as a viable candidate for privacy-critical digital image pipelines.

## 1. Introduction

The rapid advancement of digital networks has made the transmission of images faster and more widespread than ever before. However, this convenience comes with some issues—digital images often contain confidential or sensitive information and their exposure to unauthorized access, interception or tampering during transmission can lead to severe security and privacy breaches. Ensuring the protection of such data has thus become an important area of research. Traditional encryption methods, while effective for text, are often inefficient for images due to their large size,

\* Corresponding author

E-mail address: [issahzab@uds.edu.gh](mailto:issahzab@uds.edu.gh)

<https://doi.org/10.37934/ard.143.1.84108>

high redundancy and pixel correlation. Recent approaches, such as chaos-based encryption and lightweight algorithms, have improved security but still face challenges in balancing robustness, speed and resistance to advanced attacks. Machine learning (ML) has emerged as a promising alternative, yet current ML-driven encryption schemes lack efficient mechanisms for dynamic key generation and secure spatial shuffling [1,2]. This gap necessitates a novel approach that combines ensemble learning for adaptive key generation with advanced pixel disruption techniques, such as spiral ripple shuffling, to achieve both high security and computational efficiency.

The existing Chaos-based encryption techniques with their known desired features such as sensitivity to initial parameters and good randomness traits have made it one of the best solutions for securing images. Multi-map frameworks that combine several chaotic sequences to achieve confusion and diffusion in a one layered configuration have also been explored by researchers. Researchers like Güvenoğlu [3] who proposed an encryption scheme that uses multiple one-dimensional map to improve speed and security of encryption. Lai *et al.*, [4] also used a collision parity mechanism to enhance a hyperchaotic map to encrypt images in low memory resource environments. Their approach was successful with good cipher properties and low overhead, proving how chaos-based systems can be streamlined for efficiency and security. However, while chaos-based schemes exhibit attractive theoretical properties and can provide adequate security for niche applications, their practical deployment is hindered by finite-precision artefacts, insufficient cryptanalytic evaluation and operational complexities that established block ciphers have largely overcome.

Substitution boxes (S-boxes) are also another method efficient in disruption of the correlations in cypher images. In this regard, Ibrahim *et al.*, [5] proposed an encryption scheme that a custom built  $12 \times 12$  S-box for 12-bit chaos-based encryption for 12-bit medical images. Their approach was lossless and achieved a good encryption throughput when compared to the earlier 8-bit scheme. However, the 12-bit chaos-derived S-box improves throughput and accommodates higher-precision imagery, its deterministic generation, moderate non-linearity, memory cost and incomplete cryptanalytic vetting limit its robustness and hinder adoption in resource-constrained or compliance-sensitive medical environments. Future work should investigate dynamically keyed or provably optimal S-box designs, incorporate comprehensive higher-order differential and algebraic resistance testing and benchmark against lightweight block ciphers under realistic hardware profiles.

Additionally, the rapid growth in the use of digital images in the medical field has made it necessary to research into hybrid approaches that combine traditional chaos systems with machine learning approaches. An end-to-end medical image encryption scheme was proposed by Long *et al.*, [6]. Their system was based on deep learning feature encoding and decoding making use of a chaotic key generator and neural network-based diffusion. Their proposed scheme performed well, striking a delicate balance between efficient decryption and security even though there were network perturbations like noise and cropping. Also, Yasser *et al.*, [7] proposed a medical imaging specific encryption scheme that is based on chaos. Their approach achieved high computation efficiency and was resistant to various statistical attacks. However, the approach has the same limitations as in finite-precision and parameter-estimation of classical chaotic ciphers while adding the opacity and maintenance burdens of deep networks. Robust deployment will demand:

- i. Adversarial-robust training with certified error bounds
- ii. Lightweight but formally analysed post-quantum diffusion layers
- iii. Automated model-drift detection pipelines
- iv. Rigorous side-channel and compliance auditing before clinical integration can be justified.

The main contributions of this paper are as follows:

- i. We developed an ensemble-learning-driven PRNG (LogReg + RF + SVM) integrated with a Spiral Ripple Shuffle (SRS) and XOR diffusion, creating a fully key-dependent, two-stage encryption framework with linear-time complexity  $O(N)$ .
- ii. We then introduced SRS as a key-dependent permutation technique based on pixel-sum profiles, thereby enhancing nonlinear chaos and reducing vulnerability to reverse-engineering.
- iii. Consequently, we conducted detailed security and efficiency analyses, demonstrating superior performance of the proposed framework compared to recent state-of-the-art image encryption methods.

Researchers have directed attention to IoT's resource-constrained environment, leading to specialized encryption frameworks. One such approach employs a novel "Random Strip Peeling" (RSP) mechanism combined with the use of Tent and Logistic one-dimensional chaotic maps [8]. The encryption algorithm is lightweight and ensures that computational overhead remains within acceptable limits for low-power devices. Empirical results, including NPCR, UACI and entropy measures, confirm its ability to resist basic statistical attacks. Nonetheless, the discussion on advanced adversarial models is curtailed, suggesting a possible extension for broader cryptanalysis. Notwithstanding this, an explicit focus on IoT offers practical advantages for real-world deployments.

To extend beyond the baseline of standard encryption, another investigation introduces colour image encryption by leveraging multiple chaotic maps, including a KAA map, in synergy with logistic-sine and Bernoulli/Tent maps for bit-level confusion [9]. Their multi-key methodology is shown to be highly effective in scrambling pixel data and maintaining a strong confusion–diffusion link. However, while security improves through multi-key reliance, implementing and managing such a multi-tier key distribution system can be non-trivial on large-scale deployments. Overall, the authors illustrate improvements over many preceding techniques, albeit with the caveat of potential overhead in complexity.

The significance of protecting medical images is acutely highlighted by emerging deep learning methods. One illustrative example applies a Cycle GAN to encrypt images without requiring paired training datasets [10]. The resulting transformations ensure that sensitive information is thoroughly disguised, safeguarding patient privacy while also preserving diagnostic utility. This approach, however, incurs computational demands, as neural network training typically requires extensive hardware resources and carefully curated datasets. The method's higher nonlinearity can mitigate the pitfalls of simpler chaotic systems, though an overreliance on training data indicates a necessity for domain-focused model adaptation. However, this method contains high computational complexities, due to extensive hardware requirements associated with neural network training and highly specialized datasets. Higher nonlinearity of this approach possibly reduces the limitations in simpler chaotic schemes.

Combining chaotic sequences with wavelet transforms has drawn attention as well. One particular technique makes use of discrete wavelet analysis in tandem with logistic-map-based permutations, thereby introducing multi-resolution encryption [11]. This fusion yield satisfactory entropy and correlation metrics, illustrating how frequency-domain manipulations supplement classic spatial encryption. Proper selection of transform types and decomposition levels remains critical, as suboptimal parameters risk weakening the overall cryptographic strength or inflating processing costs.

In a separate thread, speed becomes a focal point, particularly for scenarios demanding real-time or near-real-time performance. A scheme that leverages row/column permutations *via* logistic maps, augmented by straightforward XOR or S-box substitutions, is illustrated to surpass classical methods in terms of encryption throughput [12]. Although results confirm robust correlation reduction and improved randomness, the paper's exploration of large-scale key management remains minimal. Nonetheless, such a streamlined pipeline appears viable where encryption time is paramount.

Various authors have pursued more advanced chaotic maps. Gao [13] presents a newly formulated 2D hyperchaotic map, validated through phase portraits, Lyapunov spectra and 0–1 tests. The proposed encryption method applies row–column shuffling and repeated diffusion, capitalizing on the map's complex attractors. While multi-iteration permutations lead to heightened security, they also raise concerns about processing time for high-resolution or batch encryptions. Additionally, the depth of comparative benchmarks against contemporary approaches remains relatively contained.

An alternative perspective focuses on partial or ROI-based encryption for medical images. Prabhavathi *et al.*, [14] propose a morphological technique to extract the most salient (and diagnostic) portion of an image, followed by an enhanced zigzag transform for confusion and a 2D logistic-sine map for diffusion. This ensures computational resources are conserved by prioritizing the encryption of critical tissue regions. Yet, the onus is on accurate ROI segmentation. Should morphological operators yield suboptimal results, sensitive segments might inadvertently remain exposed or mis-encrypted.

An additional lightweight scheme for healthcare data merges multiple chaotic elements such as Henon maps, Brownian motion and Chen's chaotic system to randomly shuffle and then apply XOR-based diffusion [15]. Here, combining three separate chaotic systems extends the key space significantly. The complexity introduced, however, underscores the logistical challenge in synchronizing parameters among multiple hospital nodes. Despite such intricacies, the authors emphasize strong conformance to recognized tests, such as the NIST suite.

Gao *et al.*, [16] demonstrate a fractional order hyperchaotic system for encrypting multiple images simultaneously. By fusing multiple grayscale images into one colour image, the method conducts row–column permutations and pixel-level diffusion. This approach is intended to lower overhead when numerous images need encryption in block. While this design proves potentially beneficial for settings like multi-modal medical imaging, it could become unwieldy in continuous real-time applications or for a large volume of images, especially if partial decryption or separate usage is required.

Yousif *et al.*, [17] proposed a cryptosystem which adds DNA encoding to chaos. They performed the permutation, substitution and diffusion using an encryption process with two rounds of six steps. The long encryption time of their system makes it not a good choice for real-time applications. Similarly, Gupta *et al.*, [18] utilized DNA encoding and crisscross diffusion to design a chaotic medical image encryption scheme with improved resistance to differential attacks. A combination of 3D image encryption, compression and non-autonomous Lorenz system was presented in Singh *et al.*, [19]. Two points are noted about their system. Firstly, the encryption time is extremely long because of the employed iterative mechanism. Secondly, their scheme was not tested against differential attacks, which makes us suspect that it may not be immune to such attacks.

Finally, some researchers integrate cellular automata (specifically Rule 30), an S-box for substitution and the Lorenz system for further diffusion [20]. This multi-stage scheme demonstrates a capacity to minimize correlation among pixels across the red, green and blue channels. Although the encryption speed of about 0.61 Mbps may suffice for moderate throughput, adding multiple keys, one from the CA-based sequence, another from the Lorenz system—can complicate rekeying

protocols. Nonetheless, thoroughly tested metrics such as MSE, NPCR and global entropy validate the architecture's reliability.

Collectively, these investigations underscore the evolving nature of image encryption, where chaos theory remains integral. Whether used in basic form for lightweight IoT solutions [8] or augmented by wavelets [11] or even neural models [10], chaos-based systems persistently appear as a linchpin for unpredictability. Simultaneously, the research trajectory in medical imaging leans toward specialized region of interest (ROI) encryption [14] or advanced hyperchaotic expansions [15,16], reflecting the domain's stringent need for confidentiality and performance. Despite the progress, key distribution complexities, the computational cost of multi-layer transformations and the necessity for robust cryptanalysis under sophisticated attack models remain the focus areas for prospective refinement.

## 2 Propose Methodology

### 2.1 Overview

To achieve secured and computationally efficient communications, we propose a framework that integrates a high-entropy key generation, two-stage image encryption and Lossless decryption as well as a trained ensemble classifier (Logistic Regression, Random Forest and SVM with soft voting) to generate pseudorandom sequences which are validated by standard randomness tests. The pseudorandom sequence is used as key. The two-stage system uses a spiral ripple shuffle (SRS) for row/column permutation to disrupt spatial correlations and a bitwise XOR to achieve a strong diffusion of pixel intensities. The reverse or decryption involves applying the inverse XOR and reverse row/column shuffles to guarantee accurate plain image recovery with the correct key.

By merging machine learning for key synthesis with well-established cryptographic primitives (permutation and XOR), the system addresses both the need for unpredictable keys and importance of thorough confusion-diffusion properties in image encryption.

### 2.2 Machine-Learning–Based PRNG for Key Image Generation

A fundamental part of this research is the PRNG component, which employs an ensemble learning model to generate the key image. Below, we describe the steps for data processing, iterative training, hyperparameter optimization and final key generations. The processes involved in the PRNG design are as follows:

#### 2.2.1 Data preparation and ensemble training

- i. Data Acquisition
  - A labelled dataset (e.g., `clustered_data.csv`), containing numerical features are corresponding labels, is utilized to train a multi-class ensemble classifier
  - The dataset is split into Features ( $X$ ) and labels ( $y$ ).
- ii. Feature Scaling
  - A *MinMaxScaler* is applied to normalize the features onto the interval  $[0,1]$ . This approach standardizes input space, enhancing classifier stability.
- iii. Ensemble
  - The data set is split into training and testing data.
  - Three base models namely Logistic regression, random forest and SVM (RBF kernel) are trained.

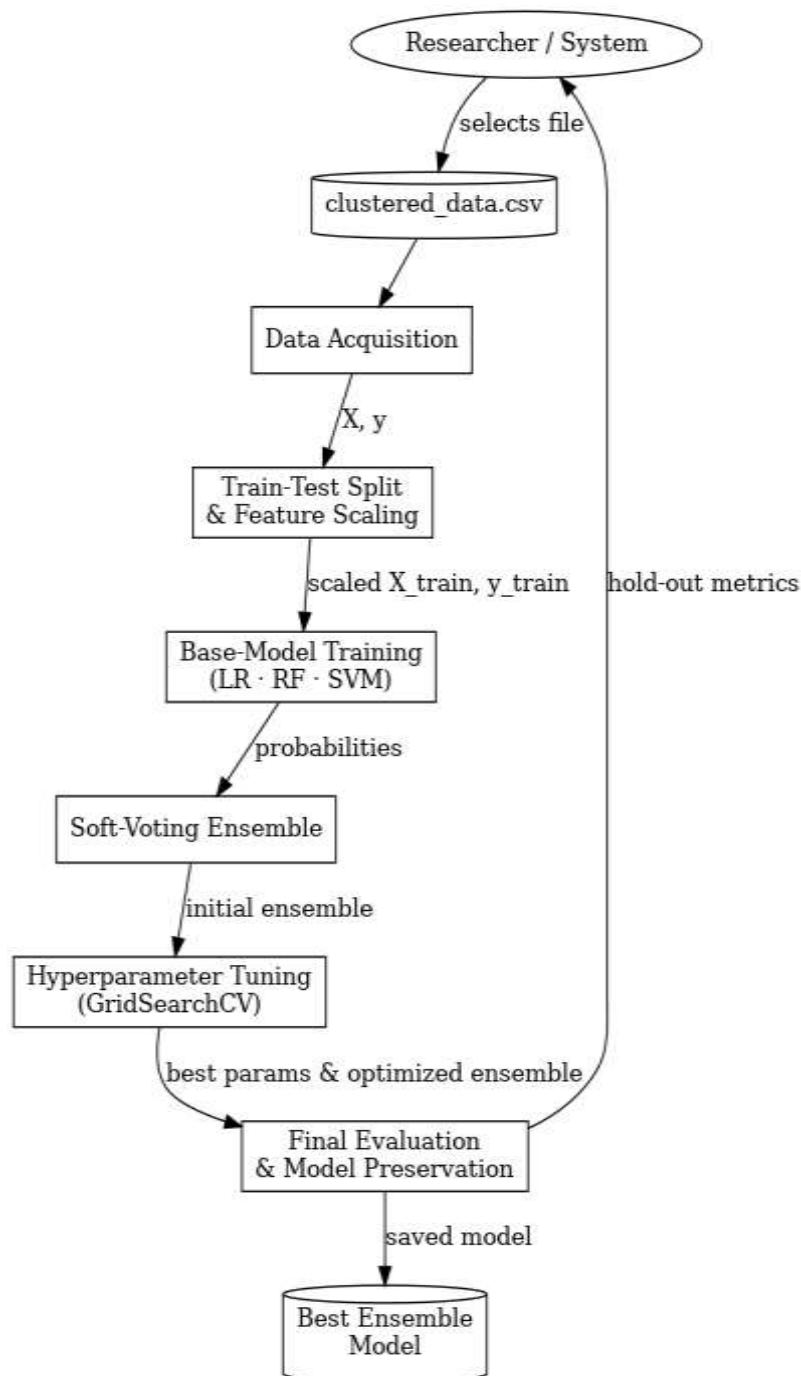
- A soft voting *VotingClassifier* is then created, aggregating and probabilistic outputs of the based models to reduce variance and improve overall predictive performance.
- iv. Hyperparameter Tuning
  - *GridSearchCV* is applied to each base learner, optimizing parameters such as  $c$  (for Logistic Regression and Support Vector Machine),  $n\_estimators$  (for Random Forest) and kernel-specific parameters ( $gamma$  for support vector machine).
  - The best-performing configurations are selected based on cross-validation accuracy and standard error bounds.
- v. Final Model Selection
  - A hold-out test set (with a fixed random state for reproducibility) is used to finalize the ensemble classifier.
  - The best ensemble model is preserved for pseudorandom sequence generation, ensuring robust performance metrics.

The data flow diagram of the data preparation and ensemble model is shown in Figure 1 below.

### 2.3 Pseudorandom Sequence Generation

Once the ensemble classifier has been validated, the model serves as a PRNG. Below are the steps involved in the sequence generation:

- i. Random Feature Inputs
  - Synthetic feature vectors are generated uniformly across the  $[0,1]$  domain or across the min-max bounds of the original data set.
  - Each vector is fed into the classifier to obtain a probability distribution over the learned classes.
- ii. Class Probability Mapping
  - A class is selected in proportion to its output probability, reflecting a stochastic choice.
  - A numeric subrange (e.g.,  $[1, 10000]$ ) is assigned to each class; one integer from the chosen class's subrange is randomly drawn. This ensures class decisions are further randomized.
- iii. Key Imation Formation
  - The validated pseudorandom sequence is reshaped into a 2D or 3D array  $K$  of dimensions  $(H, W, C)$ , matching that of the plain image  $I$ .
  - For colour encryption, multiple random sequences or channels can be stitched together for  $K$ .



**Fig. 1.** Data preparation and ensemble training model



The illustration of the key generation process is shown in Figure 2 below.

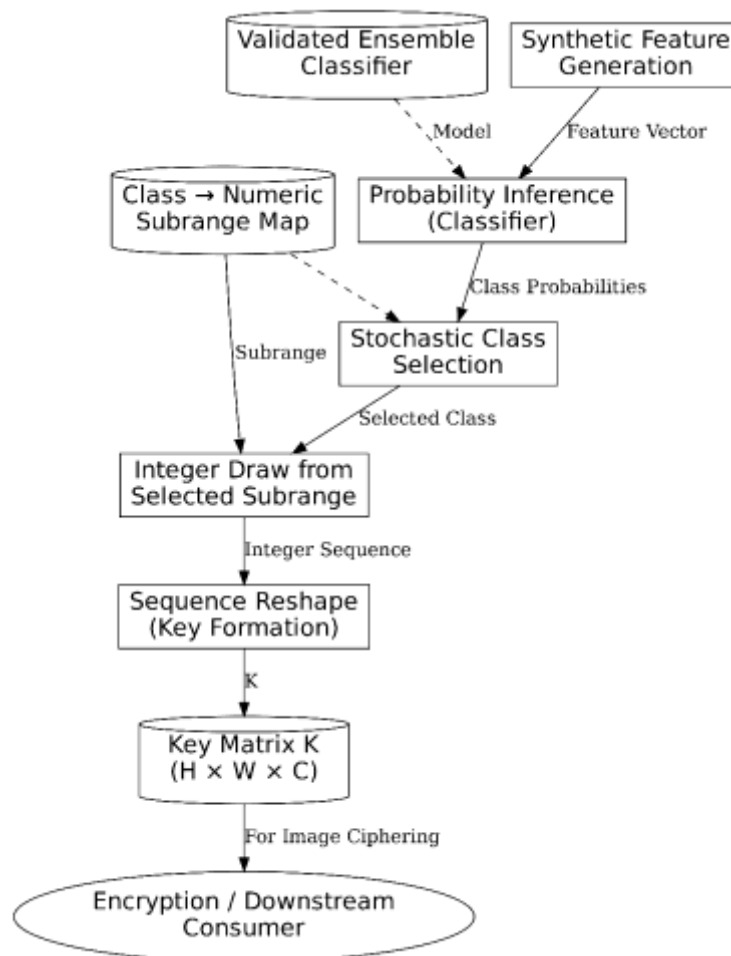


Fig. 2. Random key generation using ensemble learning

## 2.4 Spiral Ripple Shuffle (SRS)

The Spiral Ripple Shuffle (SRS) is a permutation-based approach designed to systematically disrupt the spatial relationships among pixels in an image, thereby enhancing the confusion properties in an encryption scheme. Unlike simple row- or column-only shuffles, SRS introduces a ripple-like effect by leveraging the sums of pixel intensities in the key image to dictate how rows and columns are interchanged. This integration of local pixel sums ensures that key-dependent factors guide the permutation, making it difficult for adversaries to reverse-engineer without knowing the key.

### 2.4.1 Preliminaries and notation

Let  $A \in \mathbb{Z}_{256}^{M \times N}$ ,  $S \in \mathbb{Z}_{256}^{S_r \times S_c}$ ,  $r \in \{0, 1, \dots, R - 1\}$  denote, respectively, the plaintext image, the secret seed-matrix extracted from the PRNG image and the current round index. All subscripts are taken modulo their range unless stated otherwise.



#### 2.4.2 Row-wise SRS permutation

For each row index  $i \in \{0, \dots, M-1\}$  the algorithm computes a row offset in Eq (1):

$$\Delta_r(i) = \sum_{k=0}^{s_c-1} s_i \bmod s_r, k \quad (1)$$

and swaps rows  $i$  and  $j_r(i)$  in Eq. (2):

$$j_r(i) = (i + \Delta_r(i) + r) \bmod M. \quad (2)$$

Eq. (1) and Eq. (2) define a bijective map in Eq. (3):

$$\sigma_r: i \mapsto j_r(i). \quad (3)$$

Because the implementation performs a swap for every  $i$ , the resulting permutation on the row indices can be written as the product of transpositions in Eq. (4):

$$P_r = \prod_{i=0}^{M-1} (i, \sigma_r(i)). \quad (4)$$

Each factor is its own inverse,  $(i, \sigma_r(i))^2 = id$ ; hence Eq. (5) is produced:

$$P_r^{-1} = P_r \quad (5)$$

#### 2.4.3 Column-wise SRS permutation

Analogously, Eqs. (6) to (8) are generated for each column index  $j \in \{0, \dots, N-1\}$ :

$$\Gamma_r(j) = \sum_{k=0}^{s_r-1} s_{k,j} \bmod s_c \quad (6)$$

$$i_r(j) = (j + \Gamma_r(j) + r) \bmod N \quad (7)$$

$$Q_r = \prod_{j=0}^{N-1} (j, i_r(j)), Q_r^{-1} = Q_r. \quad (8)$$

#### 2.4.4 Composite shuffle for one round

Let be the permutation realised by row shuffling followed by column shuffling in round  $r$ . Because  $P_r$  and  $Q_r$  act on disjoint index sets (rows vs. columns), they commute; nevertheless, Eq. (9) keeps the execution order of the implementation:

$$\Pi_r = Q_r \circ P_r \quad (9)$$

#### 2.4.5 Multi-round spiral ripple shuffle

For a user-selected repetition factor  $R$ , the total permutation is the composition as in Eq. (10):

$$\Pi = \Pi_{R-1} \circ \dots \circ \Pi_1 \circ \Pi_0 \quad (10)$$

Since each  $\Pi_r$  is bijective,  $\Pi$  is bijective by closure.

#### 2.4.6 Proof of correctness of reverse procedure

Proposition 1: Rows are reversed before Columns for rounds  $r = R - 1, \dots, 0$  restores  $A$ .

From Eq. (5) and Eq. (8) each elementary permutation is involutory and is shown in Eq. (11):

$$(i, \sigma_r(i)) (i + 1, \sigma_r(i + 1)) = (i, \sigma_r(i))^{-1} (i + 1, \sigma_r(i + 1))^{-1}, \quad (11)$$

The unshuffled routine applies the transpositions of  $P_r$  (or  $Q_r$ ) in reverse index order, but so ordering is immaterial: transpositions commute unless they share an index and those that share an index are identical and hence still cancel. Therefore, executing all transpositions again, regardless of order yields the identity map in Eq. (12):

$$P_r \circ P_r = id, \quad Q_r \circ Q_r = id. \quad (12)$$

The reverse pass processes the rounds in reverse order in Eq. (13):

$$\Pi^{-1} = \Pi_0^{-1} \circ \dots \circ \Pi_{R-1}^{-1} = \Pi_0 \circ \dots \circ \Pi_{R-1} = \Pi \quad (13)$$

Hence Eq. (14) is generated:

$$\Pi^{-1} \circ \Pi = id_{M \times N} \quad (14)$$

guaranteeing perfect reversibility

#### 2.5 Encryption and Decryption Process

The image encryption process proceeds in two sequential steps. SRS and Bitwise XOR aimed at thoroughly disguising the underlying structure and pixel intensity values of the original image. In the first stage, each row and column in the plaintext image is reordered based on the pixel sums extracted from a machine-learning-generated key image, thus obliterating any recognizable spatial patterns. By iterating row and column swaps multiple times, the plaintext's layout is scrambled to a high degree of confusion, effectively thwarting potential attempts at pattern-based attacks.

Following this spatial disruption, the second stage applies a bitwise XOR operation between the permuted image and the key image. Because XOR is straightforward yet cryptographically potent when paired with a high-entropy key, each pixel's intensity is shifted to a new value in a manner that is practically irreversible without the exact matching key. This final substitution stage provides an additional layer of security, ensuring that even if an adversary recognizes hints of the shuffle process, the altered pixel intensities remain indecipherable without the correct key.

Decryption is the mirror image of the spiral ripple shuffle and XOR operations. The bitwise XOR is performed first to restore the spatially shuffled pixels, followed by an inverse shuffling of columns and rows. This ensures the original image is accurately recovered if and only if the decryption process employs the same key image and permutation parameters used during encryption.

Notation and Setup:

- i. Plain Image:  $I$ , of size  $(H, W, C)$ .

- ii. Key Image:  $K$ , also of Size  $(H, W, C)$ .
- iii. Permutation Parameter:  $rep$ , controlling how many times row/columns swap repeats.  
Final Cipher Image:  $I_{enc}$  Decrypted Cipher:  $I_{dec}$  which should match the plain image  $I$ .

### 2.5.1 Encryption algorithm

#### Stage 1: Spiral Ripple shuffle (Row-Column Permutation)

- i. Row Shuffling
  - For each row index  $i \in \{0, \dots, H - 1\}$  compute  $\Delta_i = \sum(K[i \bmod H, :])$
  - Determine the new row index  $j$  via  $j = (i + \Delta_i + rep) \bmod H$ ,
  - Rows  $i$  and  $j$  of  $I$  are swapped. Repetitions of this process, controlled by  $rep$ , reinforce confusion.
- ii. Column Shuffling
  - For each column index  $j \in \{0, \dots, W - 1\}$ , compute  $\Delta_j = \sum(K[:, j \bmod W])$
  - Determine the new column index  $k$  via  $k = (j + \Delta_j + rep) \bmod W$ ,
  - Swap columns  $j$  and  $k$  in  $I$ .

This stage yields a permuted image  $I_{shuffled}$  by extensively disrupting spatial adjacency in the plain image.

#### Stage 2: Bitwise XOR (Substitution)

- i. Flattening
  - Convert  $I_{shuffled}$  and  $K$  into one-dimensional arrays of length  $H \times W \times C$ .
- ii. Pixel-wise XOR
  - For each pixel index  $p$ , compute  $I_{enc}[p] = I_{shuffled}[p] \oplus K[p]$
- iii. Cipher Image
  - The resulting array is reshaped to  $(H, W, C)$  to form the cipher image  $I_{enc}$ .

### 2.5.2 Decryption algorithm

The decryption process reverses the XOR and the row-column permutation, restoring the original plain image:

#### Stage 1: Inverse Bitwise XOR

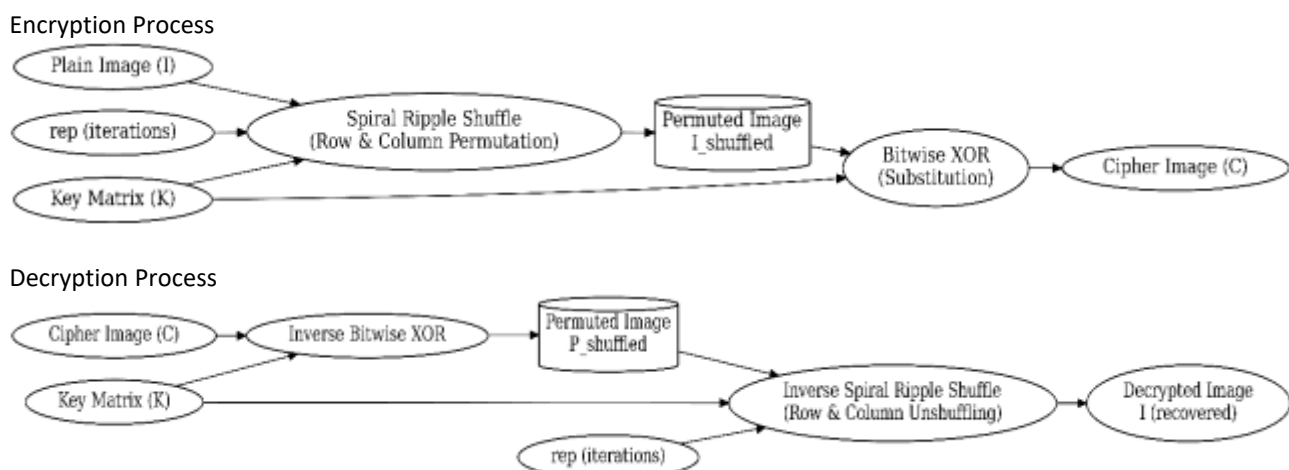
- i. Flattening
  - Convert both  $I_{enc}$  and  $K$  into 1D arrays of length  $(H \times W \times C)$
- ii. XOR Operation
  - For each index  $p$ :  $I_{shuffled}[p] = I_{enc}[p] \oplus K[p]$
  - Reshape  $I_{shuffled}$  to recover the permuted plain image.

#### Stage 2: Inverse Spiral Ripple Shuffle (Row-Column Unshuffling)

- i. Inverse Columns shuffling
  - For Each column index  $j$  from  $W - 1$  down to 0, compute  $\Delta_j = \sum(K[:, j \bmod W])$   $k = (j + \Delta_j + rep) \bmod W$ ,
  - Then swap columns  $j$  and  $k$  in  $I_{shuffled}$ .
- ii. Inverse Row Shuffling
  - For each row index  $i$  from  $H - 1$  down to 0, compute  $\Delta_i = \sum(K[i \bmod H, :])$   $m = (i + \Delta_i + rep) \bmod H$ ,
  - Then swap rows  $i$  and  $m$ .

After these unshuffling steps, the decrypted image  $I_{dec}$  should match the original plain image  $I$ , assuming the correct key  $K$  and permutation parameters are used.

The methodology described shows that, the integrated machine learning derived key generation, spiral ripple shuffle-based permutation and XOR substitution can encrypt and decrypt images. It maintains a high level of confusion and diffusion, relies on statistically verified key randomness and supports scalable implementations. By considering multiple metrics (speed, randomness, key sensitivity, cryptanalytic resistance the framework is comprehensively validated and positioned as a promising solution for secure image encryption in dynamic environments. The illustration of the encryption and decryption process is shown in Figure 3 below.



**Fig. 3.** Encryption and decryption processes

## 2.6 Evaluation Metrics

In assessing the robustness and efficiency of the proposed image encryption framework, a set of well-established evaluation metrics is employed. These metrics collectively offer insight into the security and performance of the proposed system. By examining parameters such as Number of Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI), we confirm the scheme's capacity to propagate minute key alterations across the cipher image, ensuring strong key sensitivity. Metrics like Peak Signal-to-Noise Ratio (PSNR) shed light on the quality of the recovered image and the randomness of the encryption process. Moreover, the inclusion of Digital Image Correlation (DIC) and memory/space efficiency shows the scheme's ability to obscure the original visual details without increasing overhead. Taken as a whole, these measures provide a comprehensive perspective on both the cryptographic resilience and the operational feasibility of the approach, aligning with recognized best practices in secure image processing [1].

### 2.6.1 Mean squared error (MSE)

In image encryption studies, MSE typically involves comparing the original (plain text) image with a decrypted output. A low MSE indicates that the decrypted image's intensities are nearly identical to those of the original, signifying minimal distortion or information loss. Conversely, a high MSE underscores discrepancy at the per-pixel level, hinting that the encryption–decryption pipeline either introduces noise or fails to accurately reconstruct critical details. Assuming  $X$  and  $Y$  represent two images of same dimension ( $H \times W$ ), with pixel intensities  $X_{ij}$  and  $Y_{ij}$ . The MSE is computed as in Eq. (15):

$$MSE(X, Y) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (X_{ij} - Y_{ij})^2 \quad (15)$$

MSE significance emerges from its simplicity and direct link to signal accuracy [12,15].

### 2.6.2 Peak signal-to-noise ratio (PSNR)

PSNR is typically employed in compression assessments, it is relevant here to quantify the fidelity between the decrypted image and the original. High PSNR values verify that the decryption closely reproduces the original image, indicating minimal information loss. Given the MSE between images as in Eq. (16):

$$PSNR = 10 \log_{10} \left( \frac{MAX^2}{MSE} \right) \quad (16)$$

Where,  $MAX$  is the maximum possible pixel intensity (e.g., 255 for 8-bit images) [21].

### 2.6.3 Number of pixels change rate (NPCR)

The NPCR gauges the fraction of pixels in cipher image that are altered when the key is minutely modified. It verifies whether the proposed scheme propagates key variations throughout the entire image plane. If  $D(i, j) = 1$  when pixels  $(i, j)$  differs and  $D(i, j) = 0$  otherwise, and is shown in Eq. (17):

$$NPCR = \left( \frac{\sum_{i,j} D(i,j)}{M \times N} \right) \times 100\% \quad (17)$$

Where,  $M \times N$  is the total number of pixels in the image [12].

### 2.6.4 Unified average changing intensity (UACI)

UACI assesses how much the intensity of each individual pixel is altered on average when keys differ slightly. While NPCR captures the count of changed pixels, UACI reflects the magnitude of these changes. Comparing two encrypted images  $C_1$  and  $C_2$  in Eq. (18):

$$UACI = \left[ \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{|C_1(i,j) - C_2(i,j)|}{255} \right] \times 100\% \quad (18)$$

UACI, in conjunction with NPCR, provides comprehensive insight into both the breadth and intensity of pixel modification due to slight key variations [22].

### 2.6.5 Digital image correlation (DIC)

DIC involves template matching or other alignment algorithms to ascertain how visually similar two images are. For an effectively encrypted image, DIC values between the original and the cipher should be low, reflecting minimal detectable overlap. Using normalized cross-correlation to measure the similarity between two images, let:

- i.  $T(x, y)$  be the original image,
- ii.  $I(x, y)$  be the cipher image against which the original image is compared,
- iii.  $\bar{T}$  and  $\bar{I}$  be the mean intensity of the original and cipher image respectively.

Then, the normalized cross-correlation coefficient at a given displacement  $u, v$ ) can be expressed as in Eq. (19):

$$NCC(u, v) = \frac{\sum_{x,y} [T(x,y) - \bar{T}] [I(x+u,y+v) - \bar{I}]}{\sqrt{\sum_{x,y} [T(x,y) - \bar{T}]^2 \sum_{x,y} [I(x+u,y+v) - \bar{I}]^2}} \quad (19)$$

Where, the summations  $\sum_{x,y}$  are taken over the region. This measure typically ranges from  $-1$  to  $+1$  with:

- i.  $+1$  indicating perfect positive correlation (high similarity),
- ii.  $0$  indicating no linear relationship,
- iii.  $-1$  indicating perfect negative correlation (inverse relationship).

In image encryption contexts, a DIC (NCC) value close to  $0$  (or near the lower end of the range) indicates minimal similarity, which is desirable for security because it implies the encryption has thoroughly obscured the image's original features [7].

### 2.6.6 Correlation coefficient

Correlation coefficients gauge the linear relationship among adjacent pixels. In typical plain images, neighbouring pixels are highly correlated. A strong encryption scheme drastically reduces such correlations in the cipher image, preventing adversaries from inferring spatial patterns. For any two variables  $X$  and  $Y$ , the Pearson correlation coefficient  $\rho$  is shown in Eq. (20):

$$\rho = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}} \quad (20)$$

Minimizing adjacency correlation in the encrypted image implies that the cipher obscures the images' original structure [19].

### 2.6.7 Entropy

Entropy measures uncertainty or randomness within an image's intensity distribution. An encrypted image typically exhibits high entropy, signifying greater resistance to statistical attacks. If  $p_i$  is the probability of intensity  $i$  in the image, then Eq. (21) is produced:

$$H = -\sum_{i=1}^k p_i \log_2(p_i) \quad (21)$$

Where,  $k$  is the number of intensity level (256 in an 8-bit image). Elevated entropy implies fewer discernible patterns in the cipher image, reinforcing security [9].

### 2.6.8 Structural similarity index matrix (SSIM)

SSIM is a widely recognized metric for gauging the perceptual similarity between two images, often used to assess fidelity in compression and restoration tasks. In the context of image encryption, SSIM helps compare how closely a decrypted image aligns with the original image in terms of luminance, contrast and structural content. A higher SSIM score indicates that the two images are nearly indistinguishable in human visual perception, whereas lower values reflect more visible discrepancies. Assuming  $X$  and  $Y$  are two grayscale images and  $\mu_X$  and  $\mu_Y$  are their mean intensities. The variances  $\sigma_X^2$  and  $\sigma_Y^2$  represent the spread of their intensities around the respective means and  $\sigma_{XY}$  is the covariance between them. Given constants  $C_1$  and  $C_2$  to ensure stability, the SSIM is computed as:

$$SSIM(X, Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \quad (22)$$

SSIM captures structural and perceptual similarities. In a secure encryption scenario, the SSIM between the encrypted and original images should be substantially low, signifying the cypher image retains minimal traces of the original. Similarly, the SSIM between the decrypted and original images should be relatively high, affirming successful recovery of visual details after decryption [23,24].

### 2.6.9 Bit error rate (BER)

BER originates in digital communications to measure the fraction of transmitted bits that are received incorrectly. In the context of image encryption, BER is used typically to compare the binary representations of two grayscale images, the original and an encrypted or decrypted version. A higher BER indicates more bit-level mismatches, reflecting a stronger alteration of the image data. Given two grayscale images, their intensities are flattened and unpacked into bits. Assuming  $b_1$  and  $b_2$  denote the bit streams of the first and second images, respectively, each of the length  $N$ . The BER is:

$$BER = \frac{\sum_{i=1}^N [b_1(i) \neq b_2(i)]}{N} \quad (23)$$

Where,  $[\cdot]$  is the Iverson bracket (1 if condition is true, else 0). A BER closer to 1.0 means nearly all bits diverge, implying the drastic changes. A lower BER indicates fewer bit discrepancies. BER reveals how the encryption scheme disrupts the original bit patterns.

For secure encryption, the BER between the original and encrypted images should approach 50% or higher, demonstrating that any small knowledge of the original bits confers minimal advantage in



guessing the cipher. Meanwhile, comparing the original and decrypted images allows researchers to confirm that the rightful decryption yields a low BER, confirming accurate reconstruction at the bit level [19,25].

### 3. Performance Evaluation



In this section, we first present the experimental settings. Then, the evaluation of Sensitivity and Randomness Characteristics, Entropy and Histogram Distribution, Disruption of Spatial Correlation, Image Quality and Recovery Fidelity, Digital Image Correlation and Computation Time and Practical Feasibility. Finally, thorough discussions of the results are provided.

#### 3.1 Overview of Experimental Outcomes

The proposed encryption–decryption scheme was tested on various standard images of differing sizes and characteristics, including baboon, baby, coloredChips, laure, cameraman and Lena. Table 1 lists the dimensions of each original image, along with representative cipher and decrypted outputs. A visual inspection of the cipher images shows a pronounced disruption of the original structure, while the decrypted images appear nearly indistinguishable from the original inputs. This initial examination supports the efficacy of the method’s two-stage encryption, combining SRS and bitwise XOR with a high-entropy key generated *via* the ensemble-learning–based PRNG.

**Table 1**

Plain images with their respective dimensions, ciphers and decrypted images

Image	Image Name	Image Size	Cipher Image	Decrypted Image
 <p>Original</p>	baboon.png	512 X 512	 <p>Cipher Image</p>	 <p>Recovered Image</p>
 <p>Original</p>			 <p>Cipher Image</p>	 <p>Recovered Image</p>



### 3.2 Key Sensitivity and Randomness Characteristics

A fundamental expectation in secure image encryption is that minimal changes to the key produce pronounced differences in the ciphered output. Two well-known metrics, NPCR and UACI were employed to assess this property as shown in Table 3.

#### 3.2.1 NPCR and UACI

As shown in Table 3, NPCR values exceed 99.50% for all tested images. For instance, baby.jpg and coloredChips.png exhibit NPCR values of 99.60% and 99.58%, respectively, confirming that an overwhelming fraction of the ciphered pixels flips when the key changes slightly. Equally important, UACI levels remain above 30% for each image, with baboon.png reaching 37.51% and cameraman.tif attaining 35.97%. These values highlight that not only do most pixels change, but they also shift with considerable magnitude which is a positive indication of key sensitivity and diffusion strength.


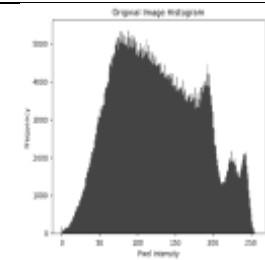
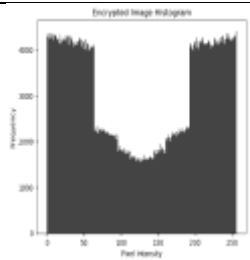
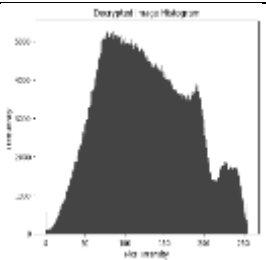

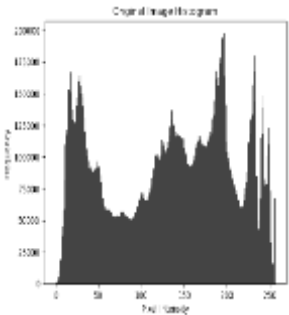
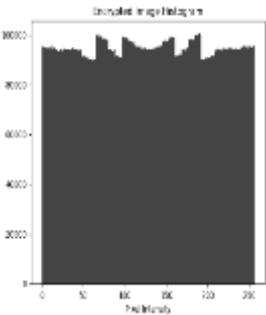
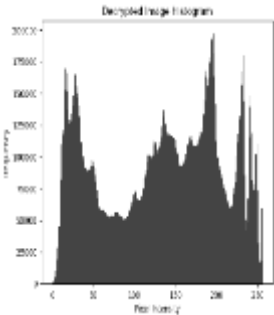

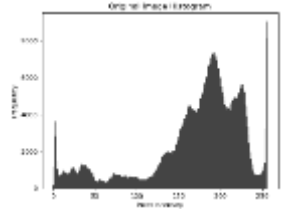
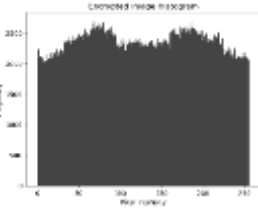


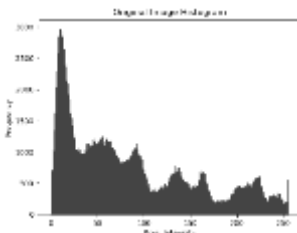
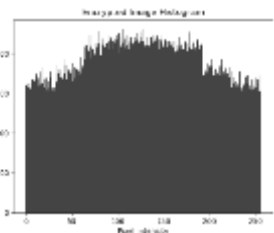
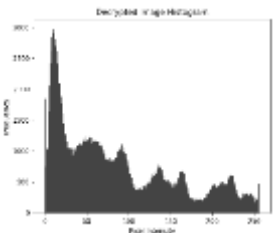
#### 3.2.2 Entropy and histogram distribution

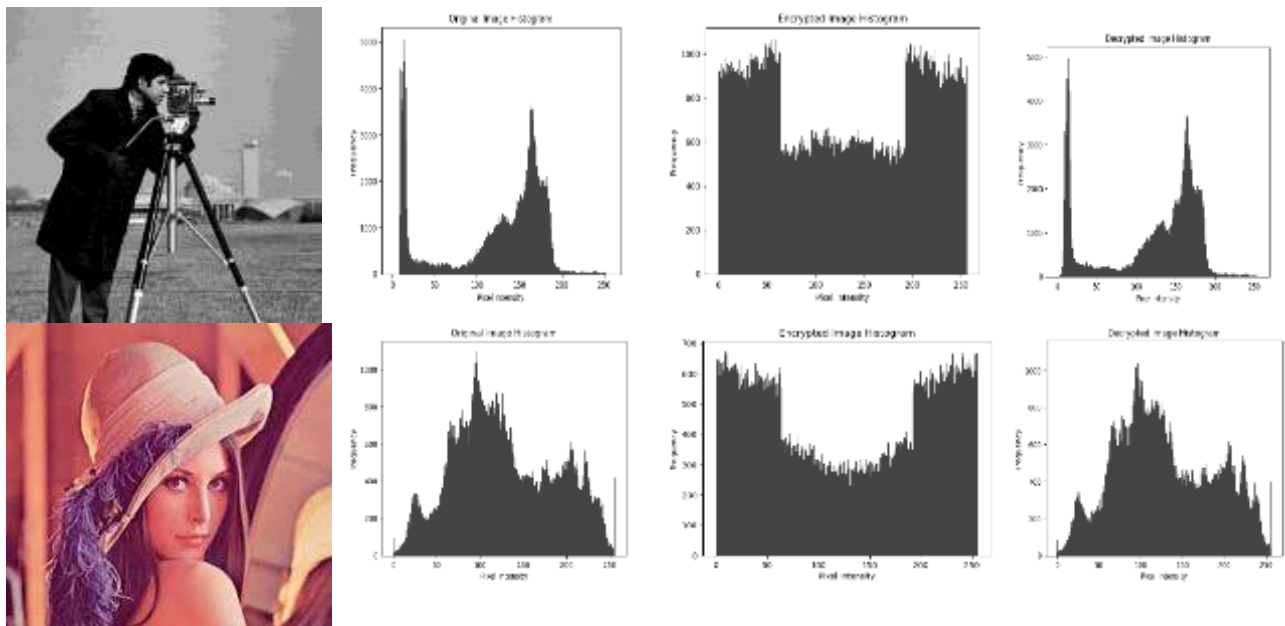
High entropy in the cipher image implies a near-uniform intensity distribution, complicating statistical or entropy-based attacks. Table 3 reveals that all cipher images exhibit entropies between

7.4745 and 7.5448, substantially elevated from the corresponding original image entropies (e.g., baboon.png's plain entropy is 5.3805, while its cipher entropy is 7.4745). These results indicate that the proposed approach effectively removes characteristic patterns from the original images.

Moreover, the cipher histograms in Table 2 approximate uniform distributions, aligning with the elevated entropy findings. The sharp divergence between plain and cipher histograms demonstrates that the encryption process scrambles pixel intensities in a manner that eludes histogram-based attacks.

**Table 2**  
Histogram of plain images, ciphers and decrypted images

Image	Histogram of Image	Histogram of Cipher	Histogram of Recovered Image
			
			
			
			



**Table 3**  
Parameters analysis of the proposed system

Image name	Original image entropy	Cipher image entropy	MSE	PSNR	NPCR (%)	UACI (%)	SSIM (Original vs encrypted)	BER (Original vs decrypted)	DIC
baboon.png	5.38	7.47	10197.00	28.85dB	99.56	37.51	0.0091	0.2147	0.0025
baby.jpg	5.46	7.54	10752.00	53.97dB	99.60	33.35	0.0066	0.0177	-0.001
coloredChips.png	5.17	7.54	10605.00	37.57dB	99.58	32.46	0.0098	0.1800	-0.001
laure.jpg	5.32	7.54	116555.00	47.56dB	99.57	31.76	0.0022	0.1872	0.0006
cameraman.tif	4.86	7.51	10319.00	44.14dB	99.58	35.97	0.0069	0.1842	-0.001
Lena.jpg	5.40	7.50	10310.34	47.63dB	99.54	36.74	0.0077	0.0560	-0.003

### 3.3 Disruption of Spatial Correlation

Natural images typically exhibit strong local correlation in horizontal, vertical and diagonal directions. Tables 4 and 5 illustrate how the proposed method drastically reduces correlation values. For instance, baby.jpg has horizontal and vertical correlation coefficients of 0.9961 and 0.9961, respectively, in the original image. After encryption, these drop to 0.3070 and 0.1095. A similar trend appears for all test images: even the cameraman.tif, which has an original vertical correlation of 0.9592, sees that figure plummet to -0.0067 upon encryption.

Such minimization of adjacency-based correlation is critical: it prevents adversaries from using spatial relationships to make inferences about underlying patterns. Scatter plots of adjacent pixels (not shown numerically here but implied by correlation analysis) confirm the near-random distribution of the ciphered pixel values.

### 3.4 Image Quality and Recovery Fidelity


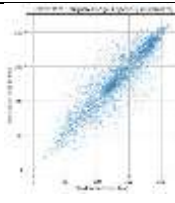
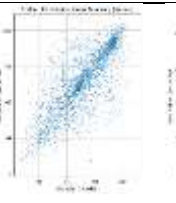
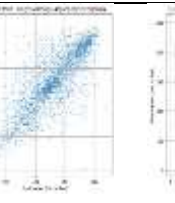
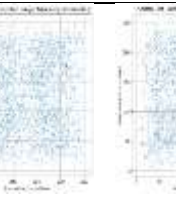
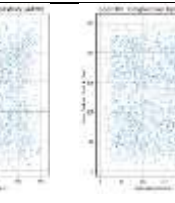


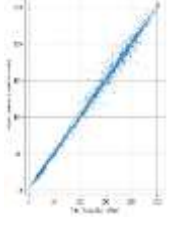
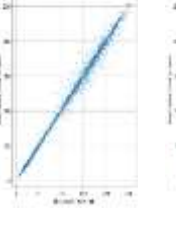
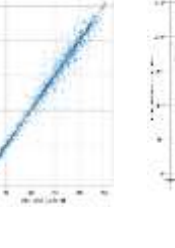
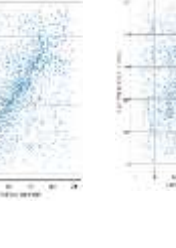
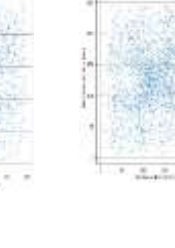


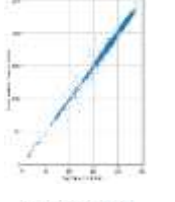
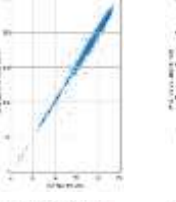
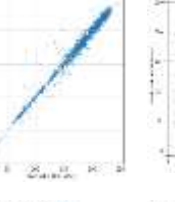
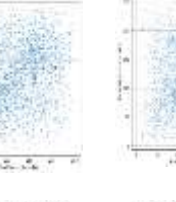
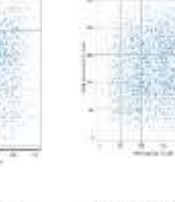


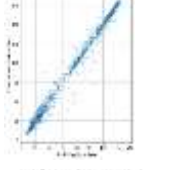
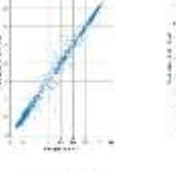
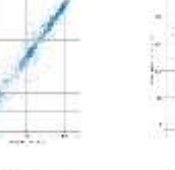
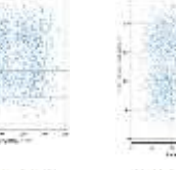
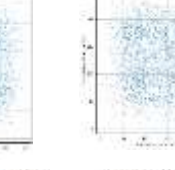


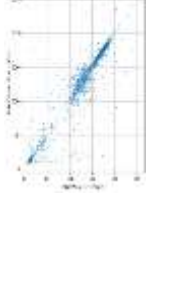
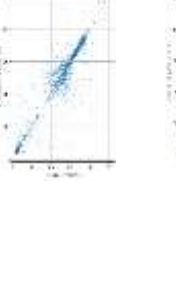
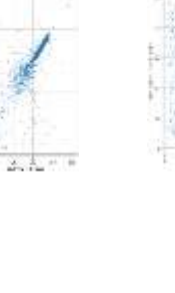
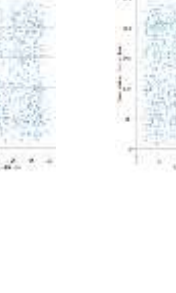
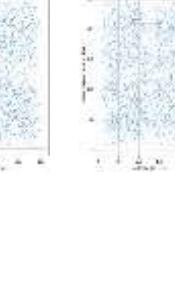


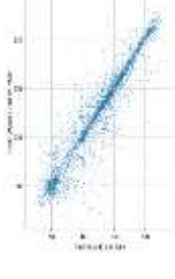
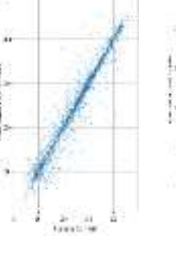
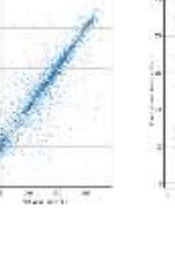
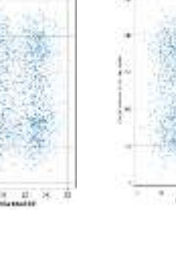
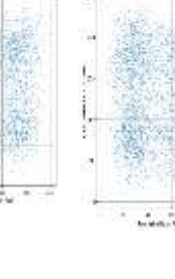

Although high security is paramount, a functional encryption scheme must also preserve the ability to accurately reconstruct the original image upon decryption. Several metrics, including MSE and PSNR in Table 3 were used to assess decryption quality. Histograms for original and decrypted



images were also compared in Table 2. The results from both Tables 2 and 3 shows that, the scheme preserves the quality of the image after reconstruction.

**Table 4**

Analysis of the correlation between the plain image and the cipher image

Direction Image	Original Images			Cipher Images		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
						
						
						
						
						
						

### 3.4.1 MSE and PSNR

As reported in Table 3, the MSE values between the original and decrypted images lie in a low range, typically on the order of  $10^{-4}$  when comparing plain vs. cipher (e.g., baboon.png's MSE is 10196.9874). These values reflect the pronounced difference between the original and encrypted images, precisely what is desired for security. The corresponding PSNR values for recovered images vary, with baby.jpg showing a relatively high PSNR of 53.97 dB, indicative of near-lossless recovery, whereas baboon.png has a more modest PSNR of 28.85 dB. In all cases, the decrypted images visually align with the originals, underscoring the consistency of the decryption process.

**Table 5**  
Analysis of the proposed system's correlations

Direction	Original Images			Cipher Images		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
baboon.png	0.8665	0.7587	0.7262	0.0305	0.0486	-0.0010
baby.jpg	0.9961	0.9961	0.9923	0.3070	0.1095	0.0458
coloredChips.png	0.9913	0.9898	0.9860	0.2821	0.2489	0.1464
laure.jpg	0.9933	0.9928	0.9851	0.0053	0.0047	0.0031
cameraman.tif	0.9335	0.9592	0.9087	-0.0065	-0.0067	0.0017
Lena.jpg	0.9274	0.9655	0.8981	0.0175	0.0246	0.0036

### 3.4.2 SSIM and BER

The Structural Similarity Index Matrix (SSIM) between the original and encrypted images remains very low (often below 0.01), indicating that the encryption thoroughly obscures perceptual details. Conversely, the Bit Error Rate (BER) reported in Table 3 underscores the difference at the bit level; for instance, coloredChips.png has a BER of 0.180006 (original vs. encrypted), showing that well over 18% of the bits deviate from the original's bit stream. This sizeable shift is advantageous from a security standpoint, as partial knowledge of the plaintext confers minimal leverage over the cipher.

### 3.5 Digital Image Correlation (DIC)

Although local correlations often capture adjacency, some advanced attacks look at larger-scale alignment via digital image correlation. The DIC values (last column of Table 3) are near zero or even negative, especially for images such as Lena.jpg (-0.0032) and baby.jpg (-0.0005). Such minimal global alignment signifies that the cipher image does not retain macrostructures of the original, adding another layer of reassurance regarding security.

### 3.6 Computation Time and Practical Feasibility

Encryption and decryption times for each tested image (Table 6) demonstrate the computational viability of the proposed scheme. Smaller images like cameraman.tif (256×256) complete encryption in approximately 0.016 seconds and decryption in about 0.016 seconds, while larger images such as baby.jpg (2250×3600) still remain under a second for encryption and just under one second for decryption on the tested hardware. These processing times indicate that, even for high-resolution inputs, the system provides rapid performance, supporting potential real-world deployment in scenarios demanding both efficiency and security.

**Table 6**

Encryption and decryption time of proposed system

Image Name	Encryption Time	Decryption Time
baboon.png	0.030862	0.031965
baby.jpg	0.938270	0.991475
coloredChips.png	0.047039	0.053560
laure.jpg	0.032043	0.015727
cameraman.tif	0.015732	0.015841
Lena.jpg	0.009106	0.005749

### 3.7 Findings and Discussions

The collective evidence from all reported metrics (NPCR, UACI, correlation coefficients, entropy, SSIM, BER and DIC) attests to the robustness of the proposed method. High NPCR/UACI underscores strong key sensitivity and diffusion properties, while elevated cipher entropy precludes straightforward statistical attacks. Markedly reduced spatial correlations confirm the system's capacity to obscure both local and global patterns and the DIC outcomes reveal negligible large-scale alignment with the original image.

Meanwhile, satisfactory PSNR and minimal distortion in the decrypted outputs confirm that the reverse process accurately recovers the original data. Consistent decryption quality across a range of test images underscores the method's adaptability. Furthermore, practical encryption and decryption times make it suitable for real-time or near-real-time applications.

Table 7 provides a quantitative comparison of several image encryption schemes, revealing notable differences in computation complexity and runtimes. Islam *et al.*, [26] report a complexity of  $O(N)^2$  with a runtime of 0.03 s, which suggests that while their approach is computationally intensive for larger images, it performs efficiently on moderate sized inputs. In contrast, Fetteha *et al.*, [27] and Ali *et al.*, [28] exhibit longer run times of approximately 0.28 s, the latter specifies a complexity of  $O(14MN + 3\log(MN))$ , indicating a more intricate process. Jain *et al.*, [29] and Erkan *et al.*, [30] do not provide explicitly complexity measures, though their NPCR and entropy values remain high. Kumari *et al.*, [31] show uniform performance across dimensions with low run time of 0.05 s while Lai *et al.*, [32] achieved a complexity of  $O(6MN + 14M + 6N)$  with a run time of 0.22 s. Notably, the scheme by Lai *et al.*, [4] operates at  $O(8MN)$  with a runtime of 0.21 s. However, our proposed scheme attains an attractive  $O(N)$  complexity and the shortest runtime of approximately 0.27 s. These findings indicate that the proposed scheme offers a promising balance between low computational complexity and robust security metrics, which is essential for real time applications and resource constrained environments.

**Table 7**

Average quantitative performance compared with existing algorithms

Schemes	Horizontal	Vertical	Diagonal	NPCR (%)	UACI (%)	Entropy	Computational Complexity	Run Time (S)
Islam <i>et al.</i> , [27]	-0.0067	0.0014	-0.0040	99.61	33.48	7.9974	$O(N^2)$	0.03
Fetteh <i>et al.</i> , [27]	1.10	-1.12	-0.49	99.65	33.41	7.9972	NA	0.28
Ali <i>et al.</i> , [28]	0.0019	0.0035	0.0008	99.62	33.46	7.9959	$O(14MN + 3\log(MN))$	0.28
Jain <i>et al.</i> , [29]	NA	NA	NA	99.61	32.88	7.7253	NA	0.56
Erkan <i>et al.</i> , [30]	0.0003	0.0002	0.0006	99.61	33.46	7.9994	$O(16MN)$	NA



Kumari <i>et al.</i> , [31]	0.0069	0.0069	0.0069	99.65	30.75	7.9967	NA	0.05
Lai <i>et al.</i> , [32]	0.000362	-0.0002	-0.00034	99.61	33.47	7.9975	$O(6MN + 14M + 6N)$	0.22
Kumar <i>et al.</i> , [33]	-0.0076	-0.0071	0.0042	99.57	33.36	7.9988	NA	0.86
Lai <i>et al.</i> , [4]	-0.0014	-0.0007	0.0003	99.62	33.44	7.008	$O(8MN)$	0.21
Proposed Scheme	0.1059833	0.0716	0.033267	99.57	34.63	7.518483	$O(N)$	0.0269564

### 3.8 Concluding Remarks on Results

In summary, the proposed scheme achieves a rigorous equilibrium between security strength and recoverability. The ensemble classifier-based key generation enhances randomness, the Spiral Ripple Shuffle disrupts spatial coherence and the bitwise XOR ensures strong diffusion. When taken together, these steps yield cipher images that are statistically indistinguishable from random noise and securely protect underlying content. At the same time, the decryption phase, employing inverse operations, reliably restores the original images without perceptible loss. These attributes, corroborated by a multifaceted evaluation, position the proposed system as a resilient and computationally viable solution for contemporary image-encryption requirements.

## 4. Conclusion

Our results show that the new image-ciphering method works very well. The scheme first lets an ensemble of three learners including Logistic Regression, Random Forest and an SVM to produce a stream of high-entropy keys. Next, a spiral-ripple shuffle breaks the usual row- and column-order of the picture and a bit-wise XOR hides the final pixel values.

Large NPCR and UACI scores prove that even a tiny key change alters almost every pixel, while the high entropy of the cipher image suggests it can withstand simple statistical or histogram attacks. Pixel-to-pixel correlations fall almost to zero, cutting the links an attacker might trace across neighbouring pixels. Finally, good PSNR values during decryption confirm that the original image is recovered without loss. Taken together, these results point to a fast, practical and secure solution for modern image protection.

From an operational perspective, the recorded encryption and decryption times remain within acceptable limits for real-time applications, indicating that the method retains computational feasibility. Such efficiency is of particular importance when large-scale or streaming image data are transmitted over unsecured channels, necessitating both rapid throughput and uncompromised cryptographic resilience.

Overall, the proposed framework stands out for its combination of machine-learning innovation and classical cryptographic rigor, offering heightened confusion and diffusion attributes without sacrificing decryption fidelity. Future enhancements may involve automated hyperparameter tuning for the ensemble-learning component, as well as optimization of the spiral ripple shuffle for images with specific structural or application-driven constraints. These refinements would further consolidate the framework's capability to handle diverse encryption scenarios while maintaining a consistently secure and reliable performance profile.

## Acknowledgement

This research was not funded by any grant.

## References

- [1] Kaur, Mandeep, Surender Singh and Manjit Kaur. "Computational image encryption techniques: a comprehensive review." *Mathematical Problems in Engineering* 2021, no. 1 (2021): 5012496. <https://doi.org/10.1155/2021/5012496>
- [2] Malsaria, Anshika, Pankaj Vyas and Manjit Kaur. "A systematic study on design and development of optical communication image encryption techniques." *Journal of Optical Communications* 45, no. s1 (2024): s1495-s1502. <https://doi.org/10.1515/joc-2023-0065>
- [3] Güvenoğlu, Erdal. "An image encryption algorithm based on multi-layered chaotic maps and its security analysis." *Connection Science* 36, no. 1 (2024): 2312108. <https://doi.org/10.1080/09540091.2024.2312108>
- [4] Lai, Qiang and Lina Ji. "A Lightweight Image Encryption Scheme Using Hyperchaotic Map and Collision-Parity Principle." *IEEE Internet of Things Journal* (2025). <https://doi.org/10.1109/JIOT.2025.3539020>
- [5] Ibrahim, Saleh, Alaa M. Abbas, Ayman A. Alharbi and Marwan Ali Albahar. "A new 12-bit chaotic image encryption scheme using a 12× 12 dynamic s-box." *IEEE Access* 12 (2024): 37631-37642. <https://doi.org/10.1109/ACCESS.2024.3374218>
- [6] Long, Bofeng, Zhong Chen, Tongzhe Liu, Ximei Wu, Chenchen He and Lujie Wang. "A novel medical image encryption scheme based on deep learning feature encoding and decoding." *IEEE Access* 12 (2024): 38382-38398. <https://doi.org/10.1109/ACCESS.2024.3371888>
- [7] Yasser, Ibrahim, Abeer T. Khalil, Mohamed A. Mohamed, Ahmed S. Samra and Fahmi Khalifa. "A robust chaos-based technique for medical image encryption." *IEEE Access* 10 (2021): 244-257. <https://doi.org/10.1109/ACCESS.2021.3138718>
- [8] İnce, Kenan, Cemile İnce and Davut Hanbay. "Random Strip Peeling: A novel lightweight image encryption for IoT devices based on colour planes permutation." *CAA Transactions on Intelligence Technology* 10, no. 2 (2025): 529-544. <https://doi.org/10.1049/cit2.12401>
- [9] Alexan, Wassim, Marwa Elkandoz, Maggie Mashaly, Eman Azab and Amr Aboshousha. "Color image encryption through chaos and kaa map." *IEEE Access* 11 (2023): 11541-11554. <https://doi.org/10.1109/ACCESS.2023.3242311>
- [10] Inam, Saba, Shamsa Kanwal, Anousha Anwar, Noor Fatima Mirza and Hessa Alfraihi. "Security of End-to-End medical images encryption system using trained deep learning encryption and decryption network." *Egyptian Informatics Journal* 28 (2024): 100541. <https://doi.org/10.1016/j.eij.2024.100541>
- [11] Pourasad, Yaghoub, Ramin Ranjbarzadeh and Abbas Mardani. "A new algorithm for digital image encryption based on chaos theory." *Entropy* 23, no. 3 (2021): 341. <https://doi.org/10.3390/e23030341>
- [12] Alghamdi, Yousef, Arslan Munir and Jawad Ahmad. "A lightweight image encryption algorithm based on chaotic map and random substitution." *Entropy* 24, no. 10 (2022): 1344. <https://doi.org/10.3390/e24101344>
- [13] Gao, Xiaohong. "Image encryption algorithm based on 2D hyperchaotic map." *Optics & Laser Technology* 142 (2021): 107252. <https://doi.org/10.1016/j.optlastec.2021.107252>
- [14] Prabhavathi, K. and M. B. Anandaraju. "Region based medical image encryption using advanced zigzag transform and 2D logistic sine map (2DLSM)." *International Journal of Cognitive Computing in Engineering* 4 (2023): 349-362. <https://doi.org/10.1016/j.ijcce.2023.10.001>
- [15] Masood, Fawad, Maha Driss, Wadii Boulila, Jawad Ahmad, Sadaqat Ur Rehman, Sana Ullah Jan, Abdullah Qayyum and William J. Buchanan. "A lightweight chaos-based medical image encryption scheme using random shuffling and XOR operations." *Wireless personal communications* 127, no. 2 (2022): 1405-1432. <https://doi.org/10.1007/s11277-021-08584-z>
- [16] Gao, Xinyu, Jiawu Yu, Santo Banerjee, Huizhen Yan and Jun Mou. "A new image encryption scheme based on fractional-order hyperchaotic system and multiple image fusion." *Scientific Reports* 11, no. 1 (2021): 15737. <https://doi.org/10.1038/s41598-021-94748-7>
- [17] Yousif, Sura F., Ali J. Abboud and Raad S. Alhumaima. "A new image encryption based on bit replacing, chaos and DNA coding techniques." *Multimedia Tools and Applications* 81, no. 19 (2022): 27453-27493. <https://doi.org/10.1007/s11042-022-12762-x>
- [18] Gupta, Mousumi, Snehashish Bhattacharjee and Biswajoy Chatterjee. "An Enhanced Security in Medical Image Encryption Based on Multi-level Chaotic DNA Diffusion." *people* 1 (2023): 8. <https://doi.org/10.18178/Joig.11.2.153-160>
- [19] Singh, Kedar Nath and Amit Kumar Singh. "Towards integrating image encryption with compression: A survey." *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 18, no. 3 (2022): 1-21. <https://doi.org/10.1145/3498342>

- [20] Alexan, Wassim, Mohamed ElBeltagy and Amr Aboshousha. "Rgb image encryption through cellular automata, s-box and the lorenz system." *Symmetry* 14, no. 3 (2022): 443. <https://doi.org/10.3390/sym14030443>
- [21] Mahendiran, N. and C. Deepa. "A comprehensive analysis on image encryption and compression techniques with the assessment of performance evaluation metrics." *SN Computer Science* 2, no. 1 (2021): 29. <https://doi.org/10.1007/s42979-020-00397-4>
- [22] SaberiKamarposhti, Morteza, Amirabbas Ghorbani and Mehdi Yadollahi. "A comprehensive survey on image encryption: Taxonomy, challenges and future directions." *Chaos, Solitons & Fractals* 178 (2024): 114361. <https://doi.org/10.1016/j.chaos.2023.114361>
- [23] Mudeng, Vicky, Minseok Kim and Se-woon Choe. "Prospects of structural similarity index for medical image analysis." *Applied Sciences* 12, no. 8 (2022): 3754. <https://doi.org/10.3390/app12083754>
- [24] Xu, Buyun, Xihai Li, Weijun Hou, Yiting Wang and Yiwei Wei. "A similarity-based ranking method for hyperspectral band selection." *IEEE Transactions on Geoscience and Remote Sensing* 59, no. 11 (2021): 9585-9599. <https://doi.org/10.1109/TGRS.2020.3048138>
- [25] Alexan, Wassim, Laila Aly, Yousef Korayem, Mohamed Gabr, Dina El-Damak, Abdallah Fathy and Hany AA Mansour. "Secure communication of military reconnaissance images over UAV-assisted relay networks." *IEEE Access* 12 (2024): 78589-78610. <https://doi.org/10.1109/ACCESS.2024.3407838>
- [26] Islam, Malik Obaid Ul and Shabir A. Parah. "Fast and lightweight image cryptosystem for IoMT applications." *Internet of Things* 25 (2024): 101083. <https://doi.org/10.1016/j.iot.2024.101083>
- [27] Fetteha, Marwan A., Wafaa S. Sayed and Lobna A. Said. "A lightweight image encryption scheme using dna coding and chaos." *Electronics* 12, no. 24 (2023): 4895. <https://doi.org/10.3390/electronics12244895>
- [28] Ali, Tahir Sajjad and Rashid Ali. "A novel color image encryption scheme based on a new dynamic compound chaotic map and S-box." *Multimedia Tools and Applications* 81, no. 15 (2022): 20585-20609. <https://doi.org/10.1007/s11042-022-12268-6>
- [29] Jain, Kurunandan, Betrant Titus, Prabhakar Krishnan, Sujitha Sudevan, P. Prabu and Ala Saleh Alluhaidan. "A lightweight multi-chaos-based image encryption Scheme for IoT Networks." *IEEE access* 12 (2024): 62118-62148. <https://doi.org/10.1109/ACCESS.2024.3377665>
- [30] Erkan, Uğur, Abdurrahim Toktas, Samet Memis, Feyza Toktas, Qiang Lai, Heping Wen and Suo Gao. "OSMRD-IE: octal-based shuffling and multi-layer rotational diffusing image encryption using 2D hybrid Michalewicz-Ackley map." *IEEE Internet of Things Journal* (2024). <https://doi.org/10.1109/JIOT.2024.3432494>
- [31] Kumari, Punam and Bhaskar Mondal. "Lightweight image encryption algorithm using NLFSR and CBC mode." *The Journal of Supercomputing* 79, no. 17 (2023): 19452-19472. <https://doi.org/10.1007/s11227-023-05415-9>
- [32] Lai, Qiang and Yuan Liu. "A cross-channel color image encryption algorithm using two-dimensional hyperchaotic map." *Expert Systems with Applications* 223 (2023): 119923. <https://doi.org/10.1016/j.eswa.2023.119923>
- [33] Kumar, Sanjay and Deepmala Sharma. "A chaotic based image encryption scheme using elliptic curve cryptography and genetic algorithm." *Artificial Intelligence Review* 57, no. 4 (2024): 87. <https://doi.org/10.1007/s10462-024-10719-0>