Evolutionary Intelligence
https://doi.org/10.1007/s12065-024-00927-6

RESEARCH PAPER

Advanced hybrid color image encryption utilizing novel chaotic neural network and 5D-hyperchaotic system

Subhashish Pal[1,3] · Jaya Mukhopadhyay[2] · Arghya Pathak[3] · Hrishikesh Mondal[4] · Mrinal Kanti Mandal[3]

Received: 13 October 2023 / Revised: 19 December 2023 / Accepted: 1 March 2024

Abstract
This paper introduces a new image encryption technique, leveraging the combined effects of a chaotic neural network and a 5D-Hyperchaotic system (HCS). The proposed method involves chaotic image matrix generation through the integration of the Ikeda map within an artificial neural network, followed by diffusion and confusion to enhance security. The 5D-HCS generates chaotic sequences using image data and a cryptographic key, which are then incorporated to produce a highly secure encrypted image. The proposed algorithm is rigorously validated through comprehensive testing like NIST suite evaluations, correlation analysis, key sensitivity assessment, and venerability to various attacks, yielding notable results such as an entropy value of 7.9992, an NPCR of 99.6, and an UACI of 33.463 for the encrypted Lena image. This study promises the method's efficacy in safeguarding sensitive visual data and positions it as an avenue for future research in image encryption as an emerging technology.

Keywords Neural network · Hyperchaotic system · Lyapunov exponents · Ikeda map · Encryption · Decryption

1 Introduction

The increasing reliance on the sharing of digital data and the widespread use of the internet have led to a growing concern about data security. Among various forms of digital data, images play a crucial role in many applications, ranging from personal photographs to medical images as well as military surveillance. Proper design of a cryptosystem can only ensure the confidentiality, integrity, and authenticity of shared images through the transmission channel. Image encryption is the process of transforming an image into its encrypted form so that only authorized parties can access the original content within it. Only efficient encryption techniques provide a solution to this problem, and the effectiveness of these techniques has been demonstrated through various studies. Conventional image encryption techniques are often used to encrypt the data transmission of images, including well-known algorithms such as the Data Encryption Standard (DES) [1] and the Advanced Encryption Standard (AES) [2]. However, in the current age of advanced computational power, the maintenance of key security and resistance against differential attacks is encountering significant obstacles. Techniques such as homomorphic encryption enable computations on encrypted images while maintaining their confidentiality;

Jaya Mukhopadhyay, Arghya Pathak, Hrishikesh Mondal and Mrinal Kanti Mandal have contributed equally to this work.

✉ Mrinal Kanti Mandal
mkmandal.phy@nitdgp.ac.in

Subhashish Pal
subhashish.pal@bcrec.ac.in

Jaya Mukhopadhyay
jaya.bhattacharjee@bcrec.ac.in

Arghya Pathak
ap.18ph1102@phd.nitdgp.ac.in

Hrishikesh Mondal
hm.13ph1505@phd.nitdgp.ac.in

1 Department of Physics, Dr. B. C. Roy Engineering College, Jemua Road, Durgapur, West Bengal 713206, India

2 Department of Mathematics, Dr. B. C. Roy Engineering College, Jemua Road, Durgapur, West Bengal 713206, India

3 Department of Physics, National Institute of Technology, Mahatma Gandhi Road, Durgapur, West Bengal 713209, India

4 Department of Physics, Durgapur Government College, Jawaharlal Nehru Avenue, Durgapur, West Bengal 713214, India

however, they often face challenges in terms of computational overhead and practicality [3]. Leveraging machine learning, some approaches deploy Generative Adversarial Networks (GANs) and chaotic systems to encrypt images in novel ways, yet they may require large datasets and intensive training [4]. For hardware-oriented solutions, FPGA-based encryption schemes have been devised to secure images using the Advanced Encryption Standard (AES), although these solutions might be constrained by hardware resources [5]. Quantum image encryption explores the intriguing possibilities of harnessing quantum properties to enhance security, but its practical implementation remains a challenging job [6–10]. Additionally, while DNA-based encryption leverages the inherent properties of DNA sequences to protect image content, its application in real-world scenarios may be limited due to the complex biological processes involved [11–14]. Along with blockchain-enhanced encryption [15, 16], chaotic maps fused with neural networks[17–20], sparse representation techniques [21, 22], and deep learning-based approaches [23, 24], these new methods are part of an exciting landscape of new image encryption methods. Despite their promises, these techniques also come with new challenges that must be addressed to fully realize their potential in reshaping the paradigm of image security and privacy in the digital age. Neural network and hyperchaos-based image encryption techniques offer promising avenues for securing digital images by leveraging the unique properties of chaos theory and neural networks [25–28]. These techniques can enhance security and computational efficiency compared to traditional encryption methods, making them suitable for a wide range of image encryption applications. The utilization of chaotic behaviour in dynamical systems and neural networks allows the generation of pseudo-random sequences that can serve as encryption keys or efficiently do the confusion-diffusion operations to enhance the security of the encrypted images. Additionally, the complex and unpredictable behaviour of the chaotic systems makes it challenging for adversaries to reverse engineer or decipher the encrypted images. However, there are several research challenges that need to be addressed in the context of NN and HCS-based image encryption. First, the security analysis of these techniques needs to be rigorously performed against various types of attacks to understand their strengths and weaknesses. This includes analysing the resistance of chaotic NN and HCS-based image encryption techniques to statistical attacks, brute force attacks, and cryptanalysis attacks. Second, optimizing the neural network architecture and HCS parameters is crucial to achieving high security and computational efficiency. Further research is needed to develop optimized chaotic NN architectures and HCS parameters that strike the right balance between security and efficiency. Additionally, efficient and secure key management mechanisms need to be developed to ensure the confidentiality and integrity of the encryption keys, and mechanisms for image authentication and integrity verification need to be incorporated to ensure the authenticity and integrity of encrypted images. Considering the above facts, we have designed a chaotic neural network by seamlessly integrating the Ikeda map with an artificial neural network. Details about the design of a chaotic neural network can be found in Sect. 2. Furthermore, our encryption algorithm incorporates a 5D-HCS, as expounded upon in Sect. 3, to enhance the security of encrypting sample images. The details of the encryption procedure are discussed in Sect. 5. Section 6 consists of the test parameters and their results, and lastly, the conclusion.

2 Chaotic neural network

A chaotic neural network is a type of artificial neural network that incorporates chaos theory into the training process. It uses the chaotic map to introduce randomness and unpredictability within the weights and biases of the network. The chaotic behaviour helps to avoid getting stuck in local minima during training and can improve the network's ability to generalize the new data set. In our case, we have implemented the Ikeda map to incorporate chaos into the training process. This network consists of 3 nodes (I) as the input layer, 30 nodes (N) as hidden layers, and 3 nodes (M) as the output layer, with the input of dimension D the same as that of the test image. The output of the hidden layer is obtained by applying the Ikeda map to the linear combination of the input data and the bias term for each hidden node. The role of the Ikeda map in this implementation of a neural network is to introduce chaos and randomness into the training process. It is a nonlinear map that generates a chaotic time series from a given input value. The bifurcation diagram and the trajectory of this map are shown in Fig. 1. In this implementation, the said map is used to transform the output of the hidden layer (basically the linear combination of the input data and bias term for each hidden node) into a more complex and chaotic representation. The Ikeda map is defined as:

$$\left.\begin{aligned} u &= 1 + c(x\cos y - y\sin x) \\ v &= c(x\sin y + y\cos x) \end{aligned}\right\} \tag{1}$$

where x and y are the inputs, c = 0.92 is the scaling parameter, and u and v are the outputs. Using the following linear transformation of the hidden layer (h), the output of the network (y) is obtained.

$$y = W_2 h + b_2 \tag{2}$$

where W_2 is a M x N dimensional weight matrix and b_2 is a M-dimensional bias vector. During training, the weights

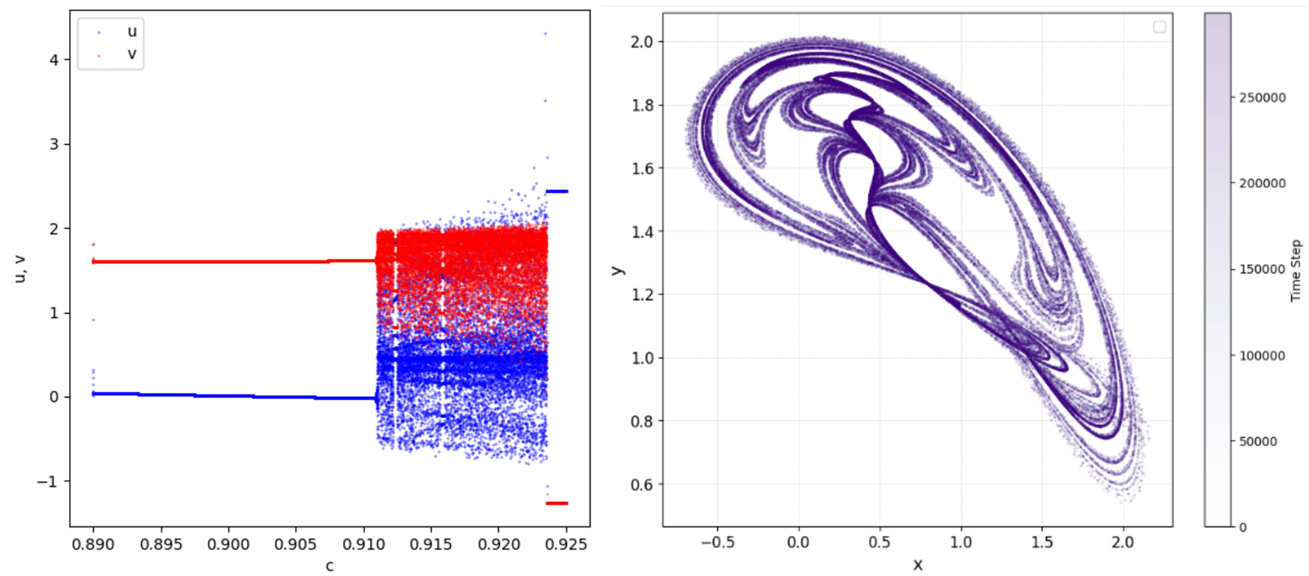

Fig. 1 Bifurcation diagram (left) and trajectory of the Ikeda map (right)

and biases are updated using back propagation with a learning rate of α. The error(e) for the output layer is defined as:

$$e = t - y \tag{3}$$

where t is the target output. The error e for the hidden layer is defined as:

$$e(i) = e\frac{\partial u}{\partial x} \tag{4}$$

where $\frac{\partial u}{\partial x}$ is the partial derivative of u with respect to x and is given by (5):

$$\frac{\partial y}{\partial x} = c(\cos y - x \sin y) \tag{5}$$

The weights and biases are updated as follows:

$$\left.\begin{array}{l} \Delta_2 = \alpha e h^T \\ W_2 = W_2 + \Delta_2 \\ b_2 = b_2 + \alpha e \end{array}\right\} \tag{6}$$

$$\left.\begin{array}{l} \Delta_1(i) = \alpha e(i) x^T \\ W_1(i) = W_1(i) + \Delta_1(i) \\ b_1(i) = b_1(i) + \alpha e(i) \end{array}\right\} \tag{7}$$

where Δ_2 is a M x N matrix, $\Delta_1(i)$ is a N x D matrix for each hidden node i, x is the input data, h is the output of the hidden layer, $W_1(i)$ is the weight matrix for hidden node i, and $b_1(i)$ is the bias term for hidden node i. In the constructor of the chaotic neural network, the weights and biases are set to random values, and the random number generator is seeded with four hash keys derived from the SHA-256 hash of the input image to make sure that the training process is random. The purpose of using the hash keys is to ensure that the weights and biases are different for each input image, which helps the network learn more effectively. The code then generates a random sequence matrix of image dimensions and applies the neural network to the data to reconstruct a chaotic image. The chaotic image is created by iterating over each pixel in the random sequence matrix and passing it as input to the network. The output of the network is then used to update the corresponding pixel in the chaotic image.

3 Hyperchaotic system

Hyperchaotic systems are those chaotic systems that have at least two positive Lyapunov exponents and are a topic of great interest in the fields of nonlinear dynamics and chaos theory. The concept of hyperchaos was first introduced by Otto Rössler in 1979 [29], and since then, many new hyperchaotic systems have been proposed. One such 5D-HCS [30] has been used in our work, and the dynamical equations of the system are formulated as

$$\left.\begin{array}{l} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - cy + v - xz \\ \dot{z} = -bz + xy \\ \dot{u} = mv \\ \dot{v} = -y - hu \end{array}\right\} \tag{8}$$

where x, y, z, u, and v are the state variables, and for a = 23, b = 3, c = 18, m = 12, and h = 4, the Lyapunov exponents of this system are $L_1 = 0.8732$, $L_2 = 0.1282$, $L_3 = -0.0013$, $L_4 = -0.5770$, and $L_5 = -8.4231$. The values of the system's Lyapunov exponents reveal important information about the nature of the system's stability and predictability. Two positive Lyapunov's exponents indicate that the system is highly sensitive to initial conditions, making its behaviour difficult to predict. Negative exponents, on the other hand, suggest that the system's behaviour will eventually converge to a stable state. The chaotic attractors of the system for different phase spaces are shown in Fig. 2.

4 Image encryption

The suggested encryption method uses chaotic sequences from the new chaotic neural network described in Sect. 2 and hyperchaotic sequences from the system described in Sect. 3. The randomness of these sequences is rigorously evaluated using the NIST SP 800-22 test, the results of which are shown in Table 1. An overview of the entire encryption procedure is illustrated below, and the flowchart of it is shown in Fig. 3.

Step 1: A color image (P) of dimension $M \times N \times 3$ is taken as input. The pixel intensity of each color channel of the image P is transformed into bytes of 8 bits. Subsequently, the SHA-256 function is used to compute the SHA-256 hash for each channel of the image.

Step 2: The 256-bit hash key generated from image P, along with the Ikeda map function, is used in a chaotic neural network to obtain the chaotic image C of dimension $M \times N \times 3$.

Step 3: A bit-wise XOR operation between the pixel values of the original image P and the corresponding pixel values of the chaotic image C present at the same location is performed. This process results in the generation of the first encrypted image 'I' of the same dimension as that of P and C.

Step 4: Iterative Block-Based Image Rotation begins with the smallest block size (S), increasing block size until it reaches the image's (P's) dimensions. In doing so, we have adopted the following steps:

a. Checking whether the image's height and width are divisible by S.
b. If yes, then reshape the image into blocks of size (S, S, 3); otherwise, apply zero padding.
c. Rotate the pixel positions within the blocks anticlockwise by 90 degrees.
d. Reshape the rotated blocks back to the original image shape.

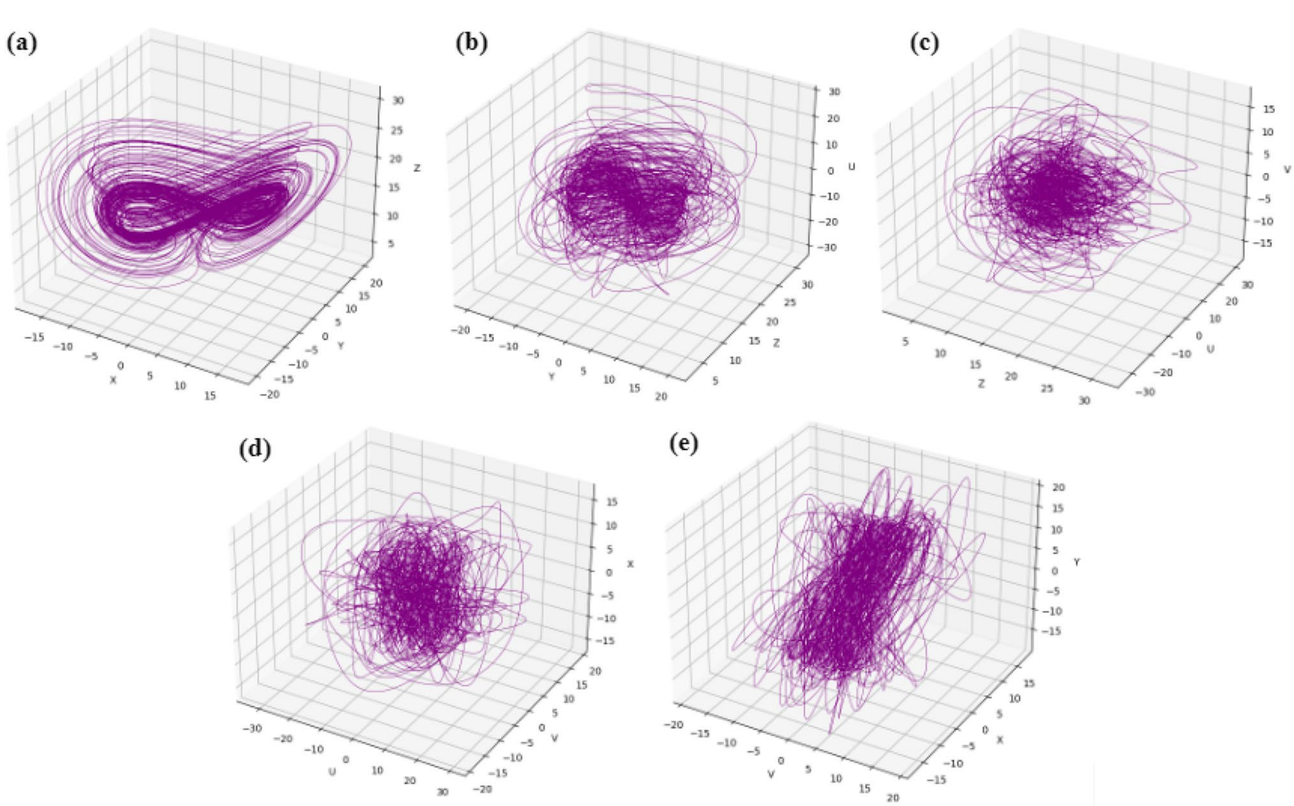

Fig. 2 Chaotic attractors of the 5D-HCS in **a** xyz, **b** uzu, **c** zuv, **d** uvx, and **e** vxy phase space

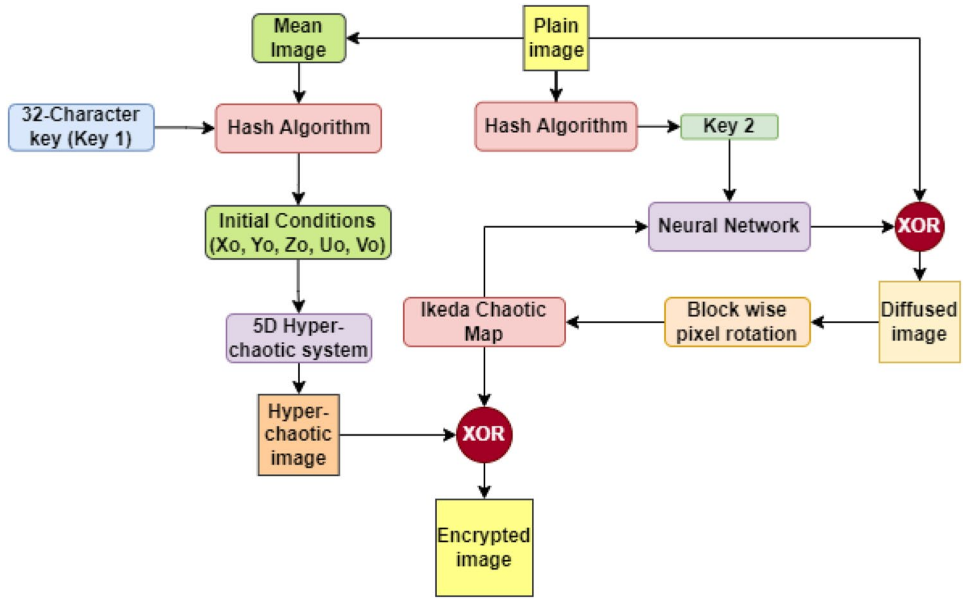

Fig. 3 Structure of the image encryption scheme

e. Updating a pixel's information for the next step of iteration by doubling S. This process repeats several times till the dimension reaches its maximum value, i.e., the image dimension M × N × 3, and finally obtains the desired image I_R.

Step 5: A diffused image $I_S\,(x_f, y_f)$ is generated by further shuffling of the pixel value of the image $I_R\,(x_i, y_i)$ by iterating the following modified equations of the Ikeda map 10 times:

$$\left.\begin{aligned} u &= 1 + d(x_i \cos bn - y_i \sin bn) \\ v &= d(x_i \sin bn - y_i \cos bn) \\ x_f &= a + cu \cos v \\ y_f &= cu \sin v \end{aligned}\right\} \tag{9}$$

Step 6: Using Eq. (8), the dynamical equations of a 5D-HCS are used to make 5 sets (k_1, k_2, k_3, k_4, and k_5) of M × N chaotic sequences. This is done after 10^7 sequences are thrown away to make the system stable. The initial values of the variables in the system equation are generated by processing the 32-bit character key with a hash function.

Step 7: Three sets of chaotic sequences of dimension M × N are generated by taking the bit XOR operation as given below.

$$k_r = k_1 \bigoplus k_2 \bigoplus k_3$$
$$k_g = k_2 \bigoplus k_3 \bigoplus k_4$$
$$k_b = k_3 \bigoplus k_4 \bigoplus k_5$$

Step 8: The final encrypted image I_E is obtained by taking the bit XOR operation of the elements of the red, green, and blue channels of I_S with k_r, k_g, and k_b, respectively.

To revert the final encrypted image back to its original form, we have implemented the inverse algorithm, depicted in Fig. 4.

The encryption and decryption procedures we introduced have undergone rigorous testing through Python code implementation. Remarkably swift, the entire process concludes within a few seconds, showcasing its efficiency and effectiveness. This is elaborated upon in Sect. 6.

5 Randomness test

The NIST SP800-22 test is a comprehensive tool for assessing image encryption pseudorandom number generator performance. This NIST test evaluates the randomness and unpredictability of a pseudorandom number generator-generated binary sequence using fifteen rigorous subtests [31]. These tests determine if a chaotic binary sequence can encrypt images securely. The NIST SP800-22 test measures sequence unpredictability using the *P*-value. *P*-values of 0.01 or higher indicate a random sequence that passes the test [32]. A *P*-value below 0.01, however, indicates a non-random sequence and fails the test. A *P*-value of 1 indicates a fully random sequence, while 0 indicates a non-random sequence. We used the NIST SP800-22 test suite to evaluate the chaotic neural network and HCS outputs for image encryption. Table 1 shows that the chaotic neural network and HCS's binary stream pass all sub-tests, proving their image encryption suitability.

6 Security and statistical analysis of the cryptosystem

This section entails conducting comprehensive experiments and security analyses utilizing Python 3.9 on a Windows 11 operating system, installed on a computer

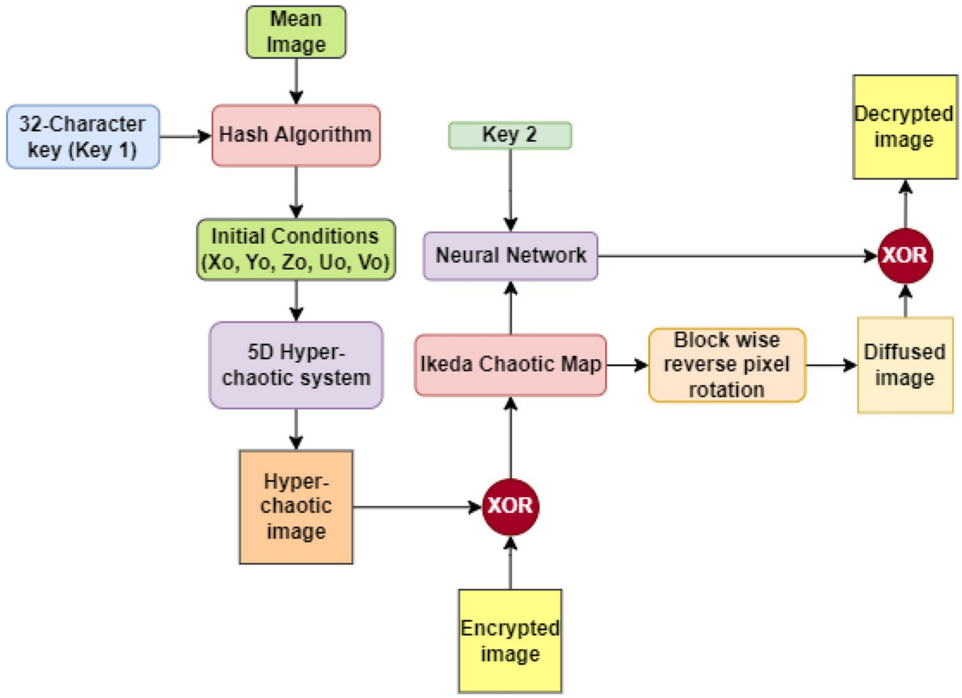

Fig. 4 Structure of the decryption scheme

Table 1 NIST test results

Statistical Test	Chaotic NN output		5D-HCS output	
	P-value	Result	*P*-value	Result
Frequency (Monobit)	0.095137	Success	0.691420	Success
Frequency (in block)	0.097551	Success	0.576690	Success
Cumulative Sums	0.150344	Success	0.756337	Success
	0.032423		0.414524	
Runs	0.890195	Success	0.620032	Success
Longest run of ones	0.118797	Success	0.241511	Success
Rank	0.262103	Success	0.605581	Success
Discrete Fourier transform (DFT)	0.743672	Success	0.027083	Success
Nonperiodic template matchings	0.463707	Success	0.434718	Success
Overlapping template matchings	0.654313	Success	0.314892	Success
Universal statistical	0.970234	Success	0.973872	Success
Approximate entropy	0.767046	Success	0.746128	Success
Random excursions	0.886128	Success	0.839246	Success
Random excursions variant	0.747654	Success	0.615842	Success
Serial	0.142092	Success	0.293375	Success
	0.478756		0.043164	
Linear complexity	0.689723	Success	0.642218	Success

furnished with an AMD Ryzen-7 5700U processor and 8.0 GB of RAM. A set of seven images of size $256 \times 256 \times 3$ have been selected from the USC-SIPI image database, which have undergone rigorous testing and are presented in Fig. 5. The purpose of selecting these images is to evaluate the robustness of the proposed image encryption scheme by encrypting them and subjecting them to various tests.

6.1 Histogram

Histogram plotting is a fundamental tool in the context of image encryption, allowing for the analysis and optimization of encryption algorithms. In image encryption, histograms depict the frequency distribution of pixel intensities in an image. The goal of image encryption is to produce ciphertext that appears random and contains no discernible patterns or structures in the histogram plot. Therefore, an encryption algorithm should produce a uniform or flat histogram that has no peaks or valleys. By analysing the histogram of encrypted images, cryptographers can detect any patterns or biases that may indicate potential weaknesses in the encryption algorithm. Furthermore, histogram analysis can be used to optimize encryption algorithms by identifying regions of an image that are particularly vulnerable to attack and adjusting the encryption process accordingly. We conducted a comprehensive analysis of the histograms for all sample images and their corresponding encrypted counterparts, out of which the histogram plot of three sample images is shown in Fig. 6. Our findings vividly demonstrate that the histogram plots of the encrypted images exhibit a significantly higher level of uniformity compared to the original images.

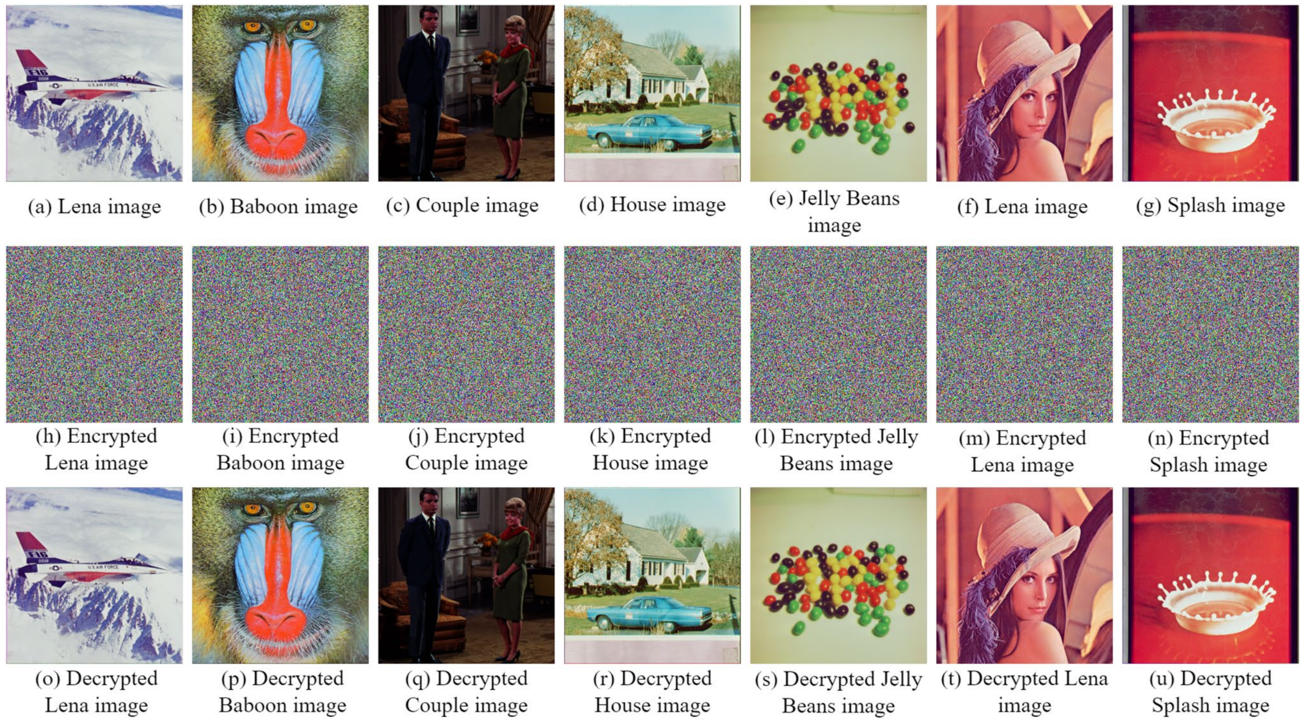

Fig. 5 Sample images and their corresponding encrypted and decrypted images

6.2 Texture analysis

Mathematical formulas that are used to measure dissimilarity (DIS), homogeneity (HOM), contrast (CON), and energy (ENE) are given by Eqs. (10), (11), (12). and (13) respectively and are commonly used to analyse image texture analytically. When comparing the original image and the encrypted image, these measures can provide valuable insights into how the

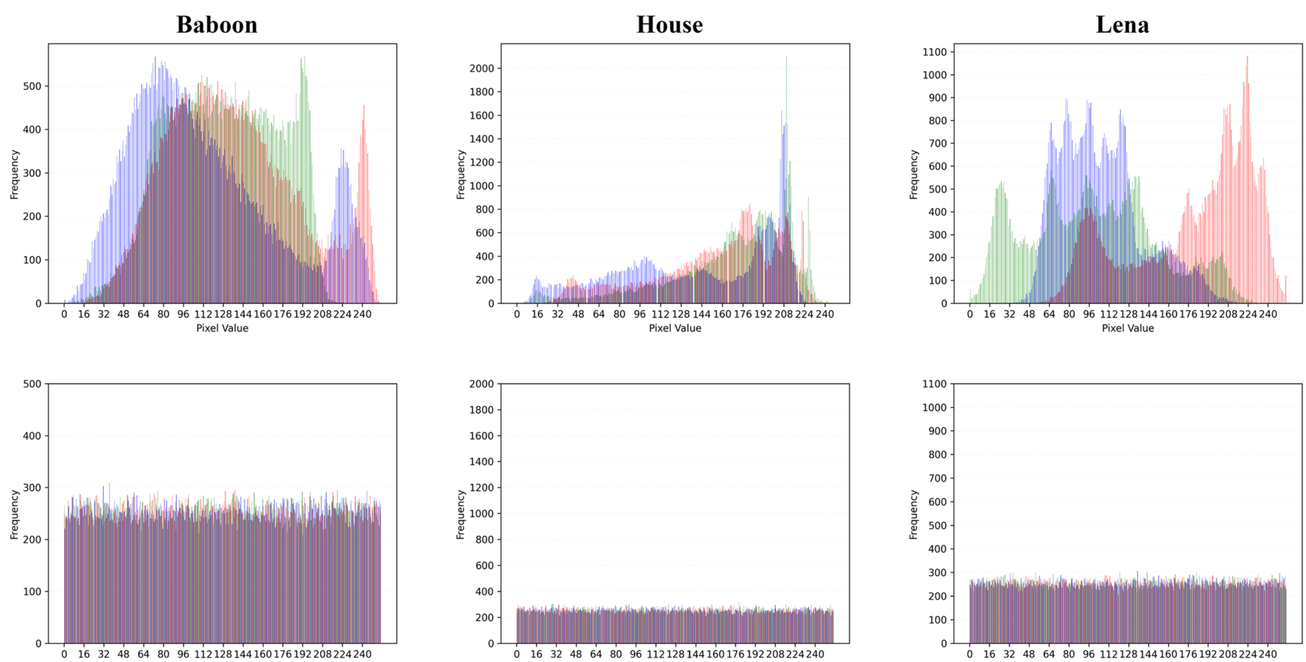

Fig. 6 Histogram plots of original image (top) and corresponding encrypted image (bottom)

texture of the image has changed. The measurement of texture is based on the grey-level co-occurrence matrix (GLCM) of the image, which describes the occurrence of pairs of pixels with a certain spatial relationship. Dissimilarity measures the degree of difference between pairs of pixels in an image. Homogeneity, on the other hand, measures the similarity of neighbouring pixels. Contrast measures the difference in intensity between neighbouring pixels. Finally, energy measures the overall uniformity of the texture. The results of texture analysis for all channels of both the original and encrypted images are displayed in Table 2. Our findings reveal notable changes: the dissimilarity and contrast values of the encrypted images have increased, while the values of homogeneity and energy have shown a significant decrease. These shifts in values underscore the enhanced robustness of the encryption algorithm.

$$DIS = \sum_{i,j=0}^{N-1} P_{i,j}|i-j| \tag{10}$$

$$HOM = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \tag{11}$$

$$CON = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2 \tag{12}$$

$$ENE = \left(\sum_{i,j=0}^{N-1} (P_{i,j}\right)^2)^{1/2} \tag{13}$$

6.3 Correlation analysis

In a 2D image, adjacent pixels tend to have a strong correlation with one another, regardless of their orientation—horizontal, vertical, or diagonal. To ensure effective encryption, it is crucial to break these pixel correlations in the original image and produce encrypted images that appear to be random noise with low levels of correlation [33, 34]. To measure the degree of correlation between pixels, correlation coefficients are calculated using specific methods, such as Eqs. (14), (15), (16). and (17).

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D_x}\sqrt{D_y}} \tag{14}$$

$$E_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{15}$$

$$D_x = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{16}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{17}$$

Here, N represents the total number of pixel pairs analysed from the test image, where x and y denote the values of adjacent pixels. Table 3 illustrates different test images for each color component, along with correlation coefficients in various orientations. Pixel correlation provides a measure of the interdependence between adjacent pixels in an image. Since a color image consists of three-color components, correlation analysis is performed separately in the horizontal, vertical, and diagonal directions for each color component. The scatter plot of correlation for all three channels is shown in Fig. 7 for the Lena image, considering both the original and encrypted images in different orientations. These scatter plots demonstrate that nearby pixels in a plain image have a strong association, while the correlation coefficient in an encrypted image rapidly decreases.

6.4 Entropy

The level of randomness in an information system can be measured quantitatively using entropy. This concept was first introduced by Shannon [35]. To determine the information entropy of a given information source (m), Eq. (18) is used:

$$E(m) = \sum_{i=0}^{2^N-1} P(m_i)\log_2 \frac{1}{P(m_i)} \tag{18}$$

Here, $P(m_i)$ represents the probability of the symbol (mi). If $E(m) = N$, the output of a source that emits 2^N symbols will be completely random. Since each symbol in our system corresponds to an 8-bit number, the ideal value for E(m) is 8, which denotes the highest level of randomness. Table 4 displays the entropy analysis results for the three-color components of the test images, showing that the entropy values are quite close to the ideal value of 8. Table 5 displays the overall information entropy of the original and encrypted images.

6.5 Key space

The suggested technique utilises the capabilities of a 5D-HCS and a chaotic neural network. The behaviour of this dynamic system is determined by five initial conditions, which confer upon it a notable susceptibility to even little modifications in its original state. Notably, five of these initial conditions are derived from the 32-characters comprising the encryption key. Once input into the 5D-HCS,

Table 2 Texture analysis results for original and encrypted images

Image	Channel	DIS	HOM	CON	ENE
Airplane	R	0.39180453	0.84343355	0.86934743	0.42346105
	G	0.58063725	0.79047329	1.62910539	0.36725482
	B	0.54666054	0.79318739	1.37662377	0.35217834
Airplane encrypted	R	5.34895833	0.16683933	42.99978554	0.06217896
	G	5.32218137	0.1672262	42.64485294	0.06210477
	B	5.31060049	0.16882532	42.53498775	0.062177
Baboon	R	1.16393995	0.57460164	2.96599265	0.13692842
	G	1.1783701	0.57320181	3.0640625	0.14887511
	B	0.96248468	0.62948601	2.21848958	0.14654955
Baboon encrypted	R	5.36971507	0.1662715	43.32501532	0.0621357
	G	5.30070466	0.16868448	42.37466299	0.06215718
	B	5.31680453	0.16757083	42.54027267	0.06214483
Couple	R	0.33262868	0.84766749	0.4901348	0.44269637
	G	0.28953738	0.87023668	0.45991115	0.43699461
	B	0.34001225	0.84899247	0.54666054	0.34589861
Couple encrypted	R	5.2928462	0.16960899	42.3136489	0.06220649
	G	5.29393382	0.16882031	42.26216299	0.06218777
	B	5.33316483	0.16688771	42.75813419	0.06221603
House	R	0.68526348	0.74093308	1.69797794	0.23227593
	G	0.68615196	0.73804778	1.64739583	0.24068622
	B	0.71132047	0.72159653	1.57838542	0.21240962
House encrypted	R	5.33509498	0.16725023	42.8930913	0.06219809
	G	5.30730699	0.16781966	42.42311581	0.0621695
	B	5.34105392	0.16740182	42.87356005	0.06217165
Jelly beans	R	0.2214614	0.90750798	0.41744792	0.37804207
	G	0.26240809	0.89726902	0.58535539	0.53912703
	B	0.21780025	0.90630086	0.37616422	0.52290837
Jelly beans encrypted	R	5.31885723	0.167517077	42.56643689	0.06215734
	G	5.33609069	0.16798348	42.84310662	0.06222251
	B	5.31778493	0.1668237	42.52973346	0.0622154
Lena	R	0.36038603	0.8414029	0.5942402	0.28471669
	G	0.39117647	0.83551308	0.74632353	0.22163925
	B	0.37449449	0.83702741	0.64557292	0.26244054
Lena encrypted	R	5.30481005	0.1699276	42.56207108	0.06218822
	G	5.29420956	0.16860198	42.27092525	0.06219422
	B	5.32573529	0.16779047	42.67913603	0.06215536
Splash	R	0.28809743	0.88746335	0.70283395	0.46022706
	G	0.35637255	0.85238297	0.76868873	0.3041013
	B	0.1629136	0.92457502	0.22792586	0.33839027
Splash encrypted	R	5.33612132	0.16731615	42.88740809	0.06216897
	G	5.31937806	0.16759906	42.61591605	0.06216972
	B	5.31479779	0.16853227	42.58223039	0.06216356

these conditions generate intricate and unpredictable chaotic sequences that serve as the foundation for our encryption process. The size of the key space, denoting the total number of possible keys, is of paramount importance in ensuring security. In our case, the key space is a staggering 2^{256}, equivalent to an astonishing 1.158×10^{77} unique keys. This colossal key space renders any brute force attack futile, as the sheer magnitude of possible keys defies computational feasibility.

6.6 Key sensitivity

The sensitivity of a cryptographic key is a pivotal aspect of a dependable encryption system, encompassing two

Table 3 Correlation coefficients for original and encrypted images in different directions

Image	Channel	H-Direction	V-Direction	D-Direction
Airplane	R	0.9345690	0.9117624	0.8850341
	G	0.9203948	0.9234021	0.8745225
	B	0.9251186	0.9201730	0.8690371
Airplane encrypted	R	0.0076107	– 0.0192912	0.0028263
	G	0.0017101	0.0029046	0.0029308
	B	– 0.0144390	0.0018197	– 0.0154175
Baboon	R	0.9059377	0.8971365	0.8564896
	G	0.8465771	0.8113468	0.7632593
	B	0.9371826	0.9118230	0.8920743
Baboon encrypted	R	– 0.0007625	0.0027948	– 0.0018353
	G	– 0.0169830	0.0031779	– 0.0067467
	B	0.0070239	– 0.0078114	0.0016794
Couple	R	0.9208100	0.9447263	0.8980410
	G	0.9346264	0.9525040	0.9119269
	B	0.9522548	0.9565993	0.9248416
Couple encrypted	R	0.0066018	– 0.0063299	0.0036854
	G	0.0016663	– 0.0027105	– 0.0016429
	B	0.0013420	– 0.0075149	– 0.0092458
House	R	0.9516401	0.9497270	0.9158934
	G	0.9017517	0.9204760	0.8523106
	B	0.9233780	0.9233068	0.8750320
House encrypted	R	– 0.0179320	– 0.0022608	0.0023506
	G	0.0043441	– 0.0012247	– 0.0018743
	B	0.0016586	0.0010722	– 0.0018984
Jelly beans	R	0.9750035	0.9776799	0.9523614
	G	0.9683100	0.9742402	0.9440366
	B	0.9704770	0.9751283	0.9461865
Jelly beans encrypted	R	0.0015510	0.0083369	0.0062765
	G	– 0.0016572	– 0.0001394	0.0029197
	B	0.0034284	0.0009420	– 0.0018277
Lena	R	0.9166174	0.9434061	0.8810268
	G	0.9424681	0.9694078	0.9160111
	B	0.9450341	0.9698716	0.9172975
Lena encrypted	R	– 0.0044834	0.0096281	– 0.0021769
	G	0.0010658	0.0022176	– 0.0003758
	B	0.0004583	– 0.0018965	– 0.0027091
Splash	R	0.9677143	0.9693644	0.9392239
	G	0.9640653	0.9783153	0.9465369
	B	0.9861851	0.9967107	0.9838430
Splash encrypted	R	– 0.0057074	– 0.0059156	0.0061765
	G	– 0.0015000	– 0.0087373	0.0010622
	B	– 0.0165272	0.0101702	– 0.0040932

critical facets. Firstly, even inconsequential changes to the key during the encryption process can have a profound impact on the resulting encrypted image. Secondly, even the slightest alteration to the key during the decryption process can prevent the accurate retrieval of the original image. To vividly illustrate this point, we conducted an experiment using the Lena image, as depicted in Fig. 8a. For the purpose of testing the variation in the encrypted image, we employed two encryption keys: “knj9t3rt5%eml@Z71!@#y5q &*90AsD2x” and “tnj9t3rt5%eml@Z71!@#y5q &*90AsD2x”, with only a single character differing. The resulting images in Fig. 8b and c exhibited a disparity of 99.63% on average across RGB channels. Figure 8d visually represents the

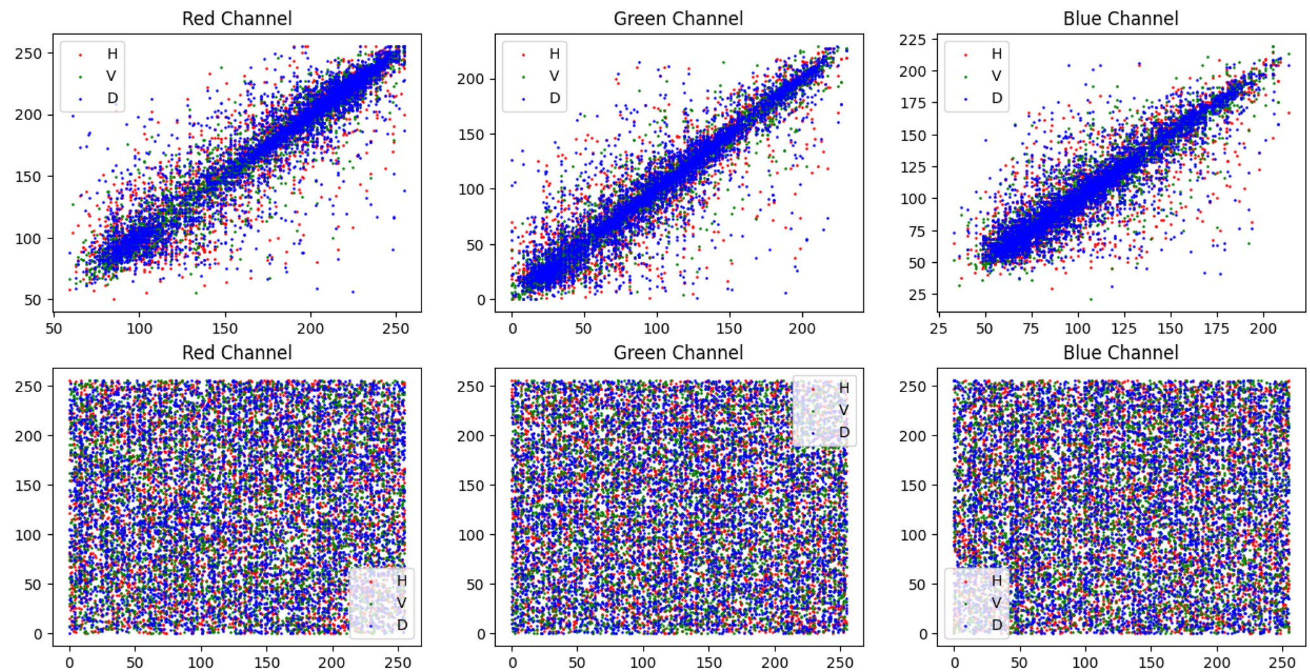

Fig. 7 Correlation scattered plot for Lena image original (top) and encrypted (bottom)

Table 4 Entropy of the original and encrypted images for different channels

Image	Original image			Encrypted image		
	R	G	B	R	G	B
Airplane	6.2207	6.8399	6.7396	7.9972	7.9974	7.9974
Baboon	7.6842	7.3827	7.6204	7.9969	7.9971	7.9972
Couple	5.9309	5.9641	6.2499	7.9972	7.9972	7.9977
House	7.4373	7.2442	7.4163	7.9970	7.9966	7.9968
Jelly beans	6.7986	6.2195	5.7919	7.9975	7.9975	7.9973
Lena	6.9716	7.5976	7.2688	7.9972	7.9977	7.9974
Splash	6.0758	6.9391	6.9507	7.9973	7.9977	7.9975

Table 5 Overall entropy of the sample images and their corresponding encrypted images

	Airplane	Baboon	Couple	House	Jelly beans	Lena	Splash
Original	6.6908	7.6947	6.2945	7.4877	6.8527	7.7508	7.2561
Encrypted	7.9990	7.9990	7.9990	7.9990	7.9991	7.9992	7.9992

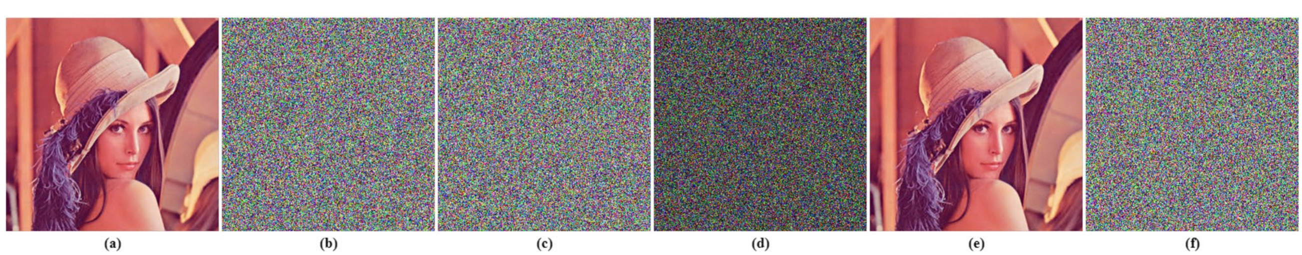

Fig. 8 Key sensitivity analysis of Lena image

image formed from the difference values of the images in Fig. 8b and c. Notably, when the encrypted image is decrypted using the same key, the original image is successfully retrieved, as demonstrated in Fig. 8e. However, even a minor discrepancy of one character in the decryption key from the original key resulted in a distinctly different image, as illustrated in Fig. 8f. This compellingly illustrates that the proposed encryption algorithm is highly sensitive to the key, with only the precise and exact key enabling successful decryption, and even the slightest deviations from the correct key yielding unrecognizable images.

6.7 Differential attack

Normalized Pixel Change Rate (NPCR) and Unified Average Changing Intensity (UACI) are important metrics used in image and video cryptography to evaluate encryption algorithms. NPCR measures the percentage of pixel value changes between the original encrypted image and the encrypted image obtained by changing one plain image pixel. Higher NPCR values, near 100, indicate greater image pixel diffusion. However, UACI measures the average intensity change in encrypted images, with a value near 33.33 indicating higher diffusion. NPCR and UACI are widely used to evaluate encryption algorithms' robustness, security, and image quality, providing valuable insights into cryptographic techniques' resilience to differential and brute-force attacks. The formulas for NPCR and UACI are:

$$NPCR = \frac{1}{L}\sum_{i,j} D(i,j) \times 100\% \tag{19}$$

$$UACI = \frac{1}{L}\sum_{i,j} \frac{|E_1(i,j) - E_2(i,j)|}{255} \times 100\% \tag{20}$$

Here, L is the total number of pixels in the image. E_1 and E_2 are the two encrypted images corresponding to the original image and original image with change in one pixel value. $D(i,j)$ is defined as:

$$D(i,j) = \begin{cases} 1 \text{ for } E_1(i,j) \neq E_2(i,j) \\ 0 \text{ for } E_1(i,j) = E_2(i,j) \end{cases} \tag{21}$$

In this experiment, a random pixel of the original image was modified and tested 10 times with one encryption round. Table 6 shows the average NPCR and UACI values. Notably, the suggested approach produces mean NPCR values above 99 percent, indicating effective pixel value diffusion. The encrypted images' UACI values are close to 33.3%, indicating a significant intensity change. These results show that the proposed image encryption method is robust and effective, supporting its use for secure data transmission and differential and brute-force protection.

6.8 Noise attack

Noise attack analysis is an important aspect of evaluating the strength and robustness of encryption and decryption algorithms, particularly in the context of encrypted images. In this study, we applied different types of noise to encrypted images, including Poisson noise with λ values 5, 15, and 30, salt and pepper noise with probabilities of 0.01, 0.05, and 0.15, and speckle noise with strengths of 0.01, 0.05, and 0.15. We obtained corresponding decrypted images, which are shown in Fig. 9, and calculated the Peak Signal-to-Noise Ratio (PSNR), and Mean Squared Error (MSE) values for both the original and decrypted images. The PSNR, and MSE data are shown in Table 7. By analysing the effects of noise on the decryption process and evaluating the PSNR, and MSE values, we were able to assess the effectiveness and robustness of the decryption algorithm. Our findings can help inform the development of more secure and reliable encryption and decryption methods to protect sensitive data in various applications.

Table 6 NPCR and UACI test results

Image	NPCR(%)			UACI(%)		
	R	G	B	R	G	B
Airplane	99.60	99.59	99.59	33.46	33.47	33.47
Baboon	99.60	99.63	99.59	33.46	33.45	33.47
Couple	99.62	99.65	99.62	33.46	33.45	33.46
House	99.60	99.59	99.60	33.46	33.47	33.46
Jelly Beans	99.60	99.61	99.58	33.46	33.46	33.47
Lena	99.61	99.59	99.60	33.46	33.47	33.46
Splash	99.57	99.62	99.62	33.48	33.46	33.46

Fig. 9 Decrypted Lena images with different noises: Salt & Pepper noise with probability 1(a) 0.01, 1(b) 0.05, and 1(c) 0.15; Speckle noise with strength 2(a) 0.01, 2(b) 0.05, and 2(c) 0.15: Poison noise with lambda value 3(a) 5, 3(b) 15, and 3(c) 30

6.9 Occlusion

In the occlusion attack analysis conducted in this study, we cropped out some portion of different dimensions of the encrypted image to simulate the effects of partial occlusion. We then obtained the corresponding decrypted images, which are shown in Fig. 10. To assess the impact of occlusion on the decryption process, we calculated the Peak Signal-to-Noise Ratio (PSNR), and Mean Squared Error (MSE) values with respect to the original image. By analysing the effects of occlusion on the decryption process and evaluating the PSNR, and MSE values, as shown in Table 8, we were able to assess the robustness and effectiveness of the decryption algorithm against occlusion attacks. Following are the equations for calculating MSE and PSNR.

$$MSE = \frac{1}{M \times N} \sum_{i,j} [O(i,j) - D(i,j)]^2 \tag{22}$$

$$PSNR = 10[\frac{I_{max}^2}{MSE}] \tag{23}$$

where, $O(i, j)$ and $D(i, j)$ represent the pixel values at the i^{th} row and j^{th} column of the original and decrypted images, respectively.

6.10 Computational complexity

The computational complexity of encryption and decryption algorithms determines cryptosystem feasibility and effectiveness. The complexity of algorithms affects processing time, memory usage, and resource needs. Encryption uses substitution, permutation, and mixing to encrypt images. Speed and resistance to attacks are often used to evaluate encryption algorithms. Decryption algorithms reverse the encryption process, so their complexity must be balanced to recover the original image easily while maintaining security. Encryption and decryption take

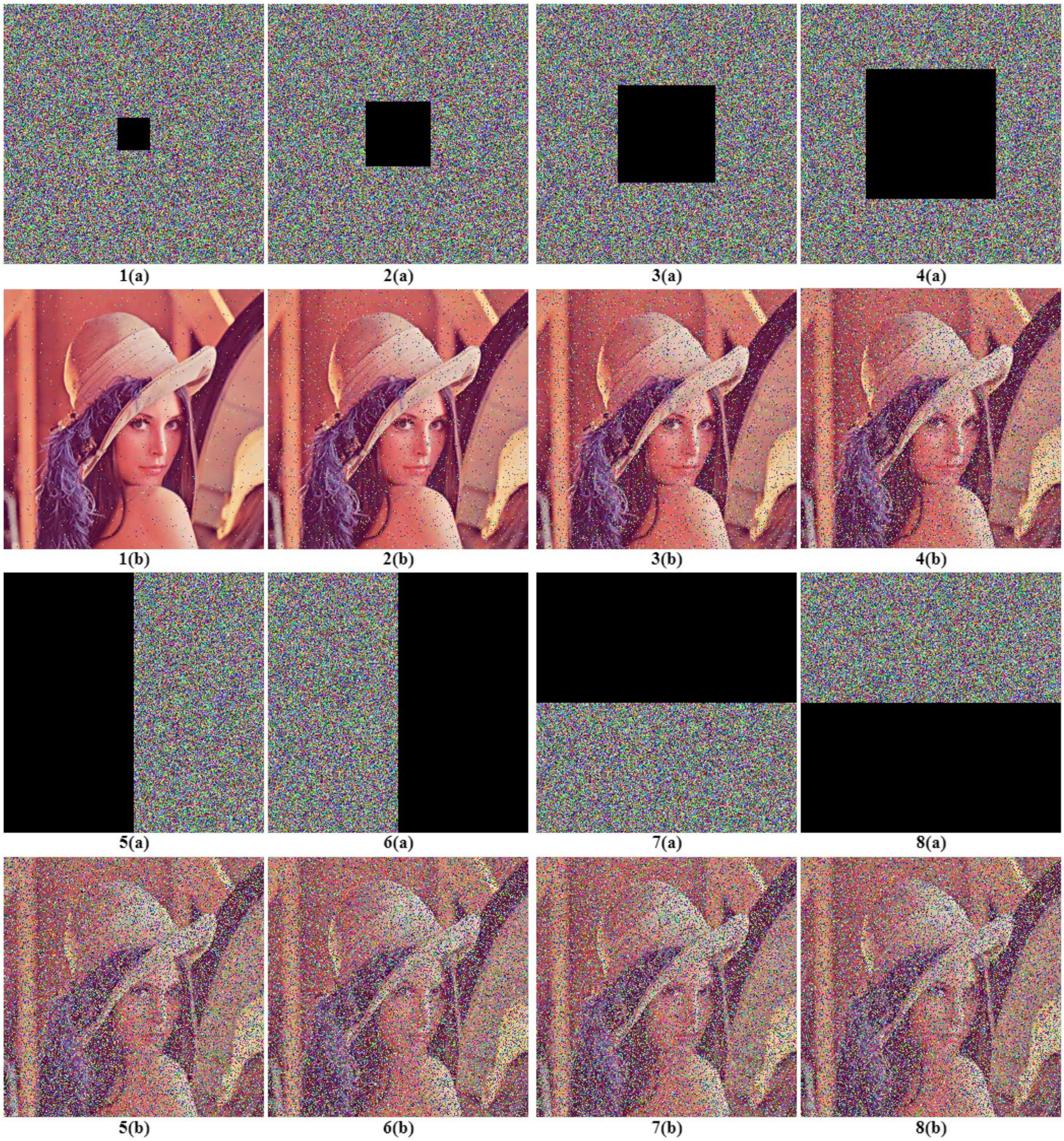

Fig. 10 Decrypted images "(b)" for occlusion size: 1(a) 32 × 32, 2(a) 64 × 64, 3(a) 96 × 96, 4(a) 128 × 128, 5(a) 50% left side, 6(a) 50% right side, 7(a) 50% top, and 8(a) 50% bottom

longer with a neural network in our cryptosystem. This trade-off requires careful evaluation to find the best balance between robust security and practical computing burden. We performed 20 encryption and decryption iterations to fully evaluate the effect. Table 9 shows the sample image average encryption and decryption times. This information helps us understand time-related factors when applying our approach in real-world situations. In a neural network-based cryptosystem, higher-configuration computers can reduce computational complexity. This could optimise processing time and performance.

7 Relative analysis

In order to evaluate the reliability and effectiveness of our suggested encryption mechanism, we performed a comparison analysis by comparing the parameter values used in our tests with those used in other current techniques that are relevant to our study. The analysis, with a special emphasis on the correlation coefficient for the Lena image, is thoroughly shown in Table 10, including the relevant results from other recent studies. Table 11

Table 7 PSNR and MSE results for different noises incorporated to Lena encrypted image

Noise Type	Parameter	PSNR (dB)	MSE
Poisson	$\lambda = 5$ (Low)	20.09	637.61
	$\lambda = 15$ (Moderate)	15.59	1795.26
	$\lambda = 30$ (High)	12.96	3290.01
Salt and Pepper	Prob. = 0.01 (Low)	28.71	87.59
	Prob. = 0.05 (Moderate)	21.66	443.45
	Prob. = 0.15 (High)	16.89	1329.66
Speckle	Strength = 0.01 (Low)	27.13	125.91
	Strength = 0.05 (Moderate)	20.15	628.01
	Strength = 0.15 (High)	15.73	1736.42

Table 8 PSNR, and MSE results for different occlusion for Lena image

Occlusion size	PSNR	MSE
32x32	26.53	144.56
64x64	20.66	558.46
96x96	14.69	2209.62
128x128	11.13	5010.64

Table 9 Encryption and decryption time for the sample images

Image	Encryption time (s)	Decryption time (s)
Airplane	12.9018	12.5179
Baboon	12.8013	12.2928
Couple	12.8271	12.3186
House	13.2781	12.8478
Jelly Beans	13.0768	12.5574
Lena	12.5923	12.1041
Splash	12.9638	12.5266

presents a comparative analysis of the information entropy of encrypted Lena in relation to some of the recently reported works. Table 12 presents a comparison of the NPCR and UACI values for the Lena image, along with those obtained from other approaches. Upon conducting a comparative analysis, it has been determined that the results obtained from our technique exhibit a significant similarity to the findings of previous studies. This outcome serves as a validation of the effectiveness and suitability of our work toward the goal of image encryption.

Table 10 Comparison of the correlation coefficient of Lena encrypted image

Methods	Channel	Direction		
		Horizontal	Vertical	Diagonal
Ref. [19]	R	− 0.0050	− 0.0096	0.0018
	G	0.0025	− 0.0032	0.0015
	B	0.0035	− 0.0023	− 0.0042
Ref. [25]	R	0.0221	0.0299	− 0.0120
	G	0.0017	0.0001	0.0265
	B	0.0073	0.0116	0.0077
Ref. [36]	R	− 0.0045	0.0149	− 0.0033
	G	0.0026	0.0126	− 0.0013
	B	− 0.000089	0.0074	0.0021
Ref. [37]	R	− 0.0073	− 0.0051	− 0.0032
	G	− 0.0022	0.0056	0.0091
	B	− 0.0172	− 0.0072	0.0003
Ref. [38]	R	− 0.0049	− 0.0174	0.0045
	G	0.0011	− 0.0156	− 0.0160
	B	− 0.0045	− 0.0175	0.0018
Ref. [39]		0.00144	− 0.00151	0.00795
Ref. [40]	R	− 0.003602	0.002683	0.001672
	G	− 0.002123	0.005589	− 0.00189
	B	− 0.00322	0.001406	− 0.000787
Ref. [41]		0.0098	0.0098	− 0.0006
Proposed	R	− 0.0044834	0.0096281	− 0.0021769
	G	0.0010658	0.0022176	− 0.0003758
	B	0.0004583	− 0.0018965	− 0.0027091

Table 11 Comparison of entropy for Lena encrypted image

Methods	R	G	B	Overall Encrypted image
Ref. [19]	7.9974	7.9975	7.9973	–
Ref. [25]	7.9994	7.9993	7.9992	–
Ref. [36]	–	–	–	7.9924
Ref. [37]	7.9970	7.9970	7.9973	–
Ref. [38]	7.9967	7.9970	7.9978	7.9990
Ref. [39]	7.9972	7.9965	7.9962	–
Ref. [40]	7.9976	7.9980	7.9981	–
Ref. [41]	–	–	–	7.9974
Proposed	7.9972	7.9977	7.9974	7.9992

8 Conclusion and future scope

The use of a chaotic neural network in conjunction with a 5D-HCS gives promising results in encrypting color images. Our proposed approach leverages the inherent complexity of a chaotic neural network to enhance the

Table 12 Comparison of NPCR and UACI values

Methods	NPCR (%)				UACI (%)			
	R	G	B	Average	R	G	B	Average
Ref. [19]	99.63	99.62	99.62	99.62	33.51	33.32	33.46	33.43
Ref. [25]	–	–	–	99.6081	–	–	–	33.4478
Ref. [36]	–	–	–	99.61	–	–	–	33.78
Ref. [37]	99.64	99.60	99.60	99.61	33.51	33.47	33.50	33.49
Ref. [38]	99.6257	99.6145	99.6257	99.6220	33.4892	33.4798	33.4916	33.4869
Ref. [39]	99.6254	99.6254	99.6254	99.6254	33.0704	30.7620	27.8720	30.6581
Ref. [40]	99.6254	99.6239	99.6330	99.6274	33.5304	33.5039	33.5343	33.5229
Ref. [41]	–	–	–	99.6281	–	–	–	33.4722
Proposed	99.61	99.59	99.60	99.60	33.46	33.47	33.46	33.463

security of the encryption process incorporated with a HCS. Through extensive experimentation and analysis, we have observed the effectiveness of this combined system in achieving robust encryption, as evidenced by the altered texture analysis parameters, histograms, and correlation coefficients. The computational complexity of this method is slightly elevated due to the incorporation of a neural network, and to mitigate this challenge, computers with higher configurations are recommended.

Though the landscape of image security continues to evolve, we hope that our approach paves the way for exciting advancements in the field of color image encryption and can be adopted to cope with the ever-changing threat landscape and fit technological innovations.

Author contributions SP: Conceptualization of problem, Python coding, Investigation, Writing of the paper. JM: Python coding and editing, Validation, Methodology, Investigation. AP and HM: Data analysis and creation of figures, paper editing. MKM: Conceptualization of problem, Investigation, Methodology, Supervision, Validation, Editing.

Declaration

Conflict of interest The authors declare that they have no Conflict of interest for the publication of the paper.

References

1. Bhanot R, Hans R (2015) A review and comparative analysis of various encryption algorithms. Int J Secur Appl 9(4):289–306
2. Dworkin MJ, Barker EB, Nechvatal JR, Foti J, Bassham LE, Roback E, Dray Jr JF (2001) Advanced encryption standard (AES)
3. Alaya B, Laouamer L, Msilini N (2020) Homomorphic encryption systems statement: trends and challenges. Comput Sci Rev 36:100235
4. Fang P, Liu H, Chengmao W, Liu M (2022) A block image encryption algorithm based on a hyperchaotic system and generative adversarial networks. Multimed Tools Appl 81(15):21811–21857
5. Shende V, Kulkarni, M (2017) Fpga based hardware implementation of hybrid cryptographic algorithm for encryption and decryption. In: 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), IEEE, pp 416–419
6. Akhshani A, Akhavan A, Lim S-C, Hassan Z (2012) An image encryption scheme based on quantum logistic map. Commun Nonlinear Sci Numer Simul 17(12):4653–4661
7. Zhou R-G, Qian W, Zhang M-Q, Shen C-Y (2013) Quantum image encryption and decryption algorithms based on quantum image geometric transformations. Int J Theor Phys 52:1802–1817
8. Yu-Guang Yang J, Tian HL, Zhou Y-H, Shi W-M (2016) Novel quantum image encryption using one-dimensional quantum cellular automata. Inf Sci 345:257–270
9. Gong L-H, He X-T, Cheng S, Hua T-X, Zhou N-R (2016) Quantum image encryption algorithm based on quantum image XOR operations. Int J Theor Phys 55:3234–3250
10. Abd-El-Atty B, Iliyasu AM, Alanezi A, Abd El-latif AA (2021) Optical image encryption based on quantum walks. Opt Lasers Eng 138:106403
11. Liu H, Wang X et al (2012) Image encryption using DNA complementary rule and chaotic maps. Appl Soft Comput 12(5):1457–1466
12. Sadkhan SB, Yaseen BS (2019) DNA-based cryptanalysis: challenges, and future trends. In: 2019 2nd scientific conference of computer sciences (SCCS), IEEE, pp 24–27
13. Wan Y, Shuangquan G, Baoxiang D (2020) A new image encryption algorithm based on composite chaos and hyperchaos combined with DNA coding. Entropy 22(2):171
14. Kar M, Kumar A, Nandi D, Mandal MK (2020) Image encryption using DNA coding and hyperchaotic system. IETE Tech Rev 37(1):12–23
15. Khan PW, Byun Y (2020) A blockchain-based secure image encryption scheme for the industrial internet of things. Entropy 22(2):175
16. Qureshi A, Megías Jiménez D (2020) Blockchain-based multimedia content protection: Review and open challenges. Appl Sci 11(1):1
17. Man Z, Li J, Di X, Sheng Y, Liu Z (2021) Double image encryption algorithm based on neural network and chaos. Chaos Solitons Fractals 152:111318
18. Feng L, Jize D, Chong F, Song W (2023) Image encryption algorithm combining chaotic image encryption and convolutional neural network. Electronics 12(16):3455
19. Wang S, Peng Q, Baoxiang D (2022) Chaotic color image encryption based on 4d chaotic maps and DNA sequence. Opt Laser Technol 148:107753
20. Liu J, Zhang J, Yin S (2023) Hybrid chaotic system-oriented artificial fish swarm neural network for image encryption. Evolut Intell 16:77–87. https://doi.org/10.1007/s12065-021-00643-5

21. Ponuma R, Amutha R (2019) Image encryption using sparse coding and compressive sensing. Multidimens Syst Signal Process 30:1895–1909
22. Karmakar J, Nandi D, Mandal MK (2020) A novel hyper-chaotic image encryption with sparse-representation based compression. Multimed Tools Appl 79:28277–28300
23. Chen J, Li X-W, Wang Q-H (2019) Deep learning for improving the robustness of image encryption. IEEE Access 7:181083–181091
24. Maniyath SR, Thanikaiselvan V (2020) An efficient image encryption using deep neural network and chaotic map. Microprocess Microsyst 77:103134
25. Yanan W, Zeng J, Dong W, Li X, Qin D, Ding Q (2022) A novel color image encryption scheme based on hyperchaos and hopfield chaotic neural network. Entropy 24(10):1474
26. Wang Y, Yang F (2021) A fractional-order CNN hyperchaotic system for image encryption algorithm. Phys Scr 96(3):035209
27. Wang S, Hong L, Jiang J (2022) An image encryption scheme using a chaotic neural network and a network with multistable hyperchaos. Optik 268:169758
28. Yang F, Mou J, Cao Y, Chu R (2020) An image encryption algorithm based on BP neural network and hyperchaotic system. China Commun 17(5):21–28
29. Rössler OE (1976) An equation for continuous chaos. Phys Lett A 57(5):397–398
30. Wei Q, Niu H (2019) Analysis and circuit design of a novel 5D hyperchaotic system. Dyn Syst Control 8(2):118–128
31. Fan C, Ding Q, Tse CK (2019) Counteracting the dynamical degradation of digital chaos by applying stochastic jump of chaotic orbits. Int J Bifurc Chaos 29(08):1930023
32. Bassham III LE, Rukhin AL, Soto J, Nechvatal JR, Smid ME, Barker EB, Leigh SD, Levenson M, Vangel M, Banks DL et al (2010) Sp 800–22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology, Gaithersburg
33. Zhong Z, Chang J, Shan M, Hao B (2012) Double image encryption using double pixel scrambling and random phase encoding. Opt Commun 285(5):584–588
34. Zhang Q, Guo L, Wei X (2010) Image encryption using DNA addition combining with chaotic maps. Math Comput Model 52(11–12):2028–2035
35. Shannon C (1948) Claude shannon. Inf Theory 3:224
36. Bhat Jasra and Ayaz Hassan Moon (2022) Color image encryption and authentication using dynamic DNA encoding and hyper chaotic system. Expert Syst Appl 206:117861
37. Zhou S, Zhao Z, Wang X (2022) Novel chaotic colour image cryptosystem with deep learning. Chaos Solitons Fractals 161:112380
38. Yan S, Li L, Binxian G, Cui Yu, Wang J, Song J (2023) Design of hyperchaotic system based on multi-scroll and its encryption algorithm in color image. Integration 88:203–221
39. Alexan W, Elkandoz M, Mashaly M, Azab E, Aboshousha A (2023) Color image encryption through chaos and KAA map. IEEE Access 11:11541–11554
40. Pal S, Mahanty A, Pathak A, Karmakar J, Mondal H, Mandal M (2023) A novel image encryption technique with four stage bit-interspersing and a 4D-hyperchaotic system. ECTI Trans Comput Inf Technol (ECTI-CIT) 17(1):105–116
41. Pathak A, Mondal H, Karmakar J, Pal S, Nandi D, Mandal MK (2023) Sparse compression-based image encryption using data encryption standards rc5. IETE Tech Rev. https://doi.org/10.1080/02564602.2023.2240286