

A visually meaningful medical image encryption scheme based on image steganography and memristive Hopfield neural networks

Wei Yao(姚卫), Xiangyun Huang(黄翔云), Jianhua Xiao(肖捡花), Fei Yu(余飞), and Yichuang Sun(孙义闯)
Citation: Chin. Phys. B, 2026, 35 (**6**): 068702. DOI: 10.1088/1674-1056/ae395d
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Memristor-coupled dynamics and synchronization in two bi-neuron Hopfield neural networks

Fangyuan Li(李芳苑), Haigang Tang(唐海刚), Yunzhen Zhang(张云贞), Bocheng Bao(包伯成), Hany Hassanin, and Lianfa Bai(柏连发)
Chin. Phys. B, 2025, 34 (**12**): 128701. DOI: 10.1088/1674-1056/ae101c

A physical memristor model for Pavlovian associative memory

Jiale Lu(卢家乐), Haofeng Ran(冉皓丰), Dirui Xie(谢頔睿), Guangdong Zhou(周广东), and Xiaofang Hu(胡小方)
Chin. Phys. B, 2025, 34 (**1**): 018703. DOI: 10.1088/1674-1056/ad8b37

Dynamical behaviors in discrete memristor-coupled small-world neuronal networks

Jieyu Lu(鲁婕妤), Xiaohua Xie(谢小华), Yaping Lu(卢亚平), Yalian Wu(吴亚联), Chunlai Li(李春来), and Minglin Ma(马铭磷)
Chin. Phys. B, 2024, 33 (**4**): 048701. DOI: 10.1088/1674-1056/ad1483

Dynamical behavior of memristor-coupled heterogeneous discrete neural networks with synaptic crosstalk

Minglin Ma(马铭磷), Kangling Xiong(熊康灵), Zhijun Li(李志军), and Shaobo He(贺少波)
Chin. Phys. B, 2024, 33 (**2**): 028706. DOI: 10.1088/1674-1056/aceee9

Dynamics and synchronization in a memristor-coupled discrete heterogeneous neuron network considering noise

Xun Yan(晏询), Zhijun Li(李志军), and Chunlai Li(李春来)
Chin. Phys. B, 2024, 33 (**2**): 028705. DOI: 10.1088/1674-1056/ad062c

---

SPECIAL TOPIC — Biophysical circuits: Modeling & applications in neuroscience

A visually meaningful medical image encryption scheme based on image steganography and memristive Hopfield neural networks

Wei Yao(姚卫)[1,2], Xiangyun Huang(黄翔云)[2], Jianhua Xiao(肖捡花)[1,†], Fei Yu(余飞)[2], and Yichuang Sun(孙义闯)[3]

[1] *School of Computer and Artificial Intelligence, Huaihua University, Huaihua* 418000, *China*
[2] *School of Physics and Electronic Science, Changsha University of Science and Technology, Changsha* 410000, *China*
[3] *School of Engineering and Technology, University of Hertfordshire, Hatfield AL*10 9*AB, UK*

(Received 16 November 2025; revised manuscript received 7 January 2026; accepted manuscript online 16 January 2026)

With the advancement of telemedicine technology, the security of digital medical images has become increasingly important. To address this issue, this paper proposes a visually meaningful color medical image encryption algorithm. First, a high-dimensional chaotic sequence is generated using a memristive Hopfield neural network. Subsequently, multi-channel pixel permutation is performed based on a chaos-driven pseudo-random strategy, followed by the implementation of a double-layer diffusion mechanism integrating cellular automata and dynamic deoxyribonucleic acid (DNA) coding. Finally, a chaos-driven cross-channel least significant bit (LSB) embedding approach is adopted. Simulation experiments and security analyses demonstrate that the proposed algorithm achieves excellent encryption performance, a large key space, and strong robustness against noise and data-loss attacks, thereby effectively ensuring the secure transmission of digital medical images.

Keywords: Hopfield neural network, memristor, chaotic sequence, image encryption algorithm

PACS: 87.18.Sn, 84.30.Bv, 05.45.–a, 07.05.Pj

DOI: 10.1088/1674-1056/ae395d **CSTR:** 32038.14.CPB.ae395d

1. Introduction

Nowadays, with the emergence of big data and the rapid advancement of cloud computing, modern networks generate, transmit, process, and store vast amounts of data. Compared with text information, digital images possess the advantages of intuitive representation and large information capacity. As a result, digital images are increasingly utilized as a major form of data transmission over modern networks, making them an important carrier for network information exchange.[1] The development of fifth-generation (5G) communication technology has further facilitated the realization of numerous telemedicine applications. Through telemedicine technology, doctors are able to provide medical services remotely and utilize the Internet to transmit relevant medical images, thereby extending high-quality medical services to a wider population.[2–4] However, unlike general multimedia data, medical images usually have high resolution, large data volume, rich grayscale information, and strong spatial correlation, while simultaneously containing highly sensitive patient privacy information. As a result, medical images are directly associated with patient diagnosis and treatment decisions. Any unauthorized disclosure, tampering, or loss of medical images may lead to serious privacy breaches and even medical misjudgments. Therefore, developing secure and efficient medical image encryption schemes suitable for telemedicine and cloud-based healthcare environments has become an urgent and indispensable research topic. The image encryption method proposed in this work is designed to satisfy the stringent security requirements of medical image protection in telemedicine and cloud-based healthcare environments.

Cryptography, as an information protection technology, aims to provide effective solutions for secure information protection. For example, information encryption refers to the process of transforming sensitive information into an unintelligible form,[5] thereby preventing unauthorized users from accessing the original information unless proper decryption is performed.[6–10] Specifically, in the field of digital image information protection, the fundamental idea is to transform meaningful image information into noise-like information through encryption techniques, thereby preventing unauthorized interpretation of digital image contents.[11,12] Due to the large data volume, high redundancy, and strong pixel correlation of digital images, conventional text-oriented encryption algorithms, such as the data encryption standard (DES),[13] advanced encryption standard (AES),[14] and Rivest–Shamir–Adleman (RSA) algorithm,[15] encounter limitations when directly applied to image encryption tasks. Conventional cryptographic algorithms are not specifically designed for the intrinsic characteristics of digital images, such as large data volume, high pixel redundancy, and strong spatial correlation. As a result, directly applying these algorithms to medical image encryption often leads to low computational effi-

†Corresponding author. E-mail: xiaojianhua@hnu.edu.cn

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

ciency, insufficient diffusion capability, and vulnerability to statistical and differential attacks, making them unsuitable for real-time and large-scale medical image protection. Consequently, these algorithms are not well suited for digital image encryption. With the advancement of cryptography research and the increasing security demands of digital image data, scholars have integrated technologies from various fields, including neural networks,[16,17] deoxyribonucleic acid (DNA) coding technology,[18] cellular automata,[19] gyrator transform,[20] compressive sensing,[21] and discrete chaotic maps,[22] to develop numerous image encryption schemes.

In cryptography, chaotic systems are considered ideal pseudo-random generators due to their high sensitivity to initial states, pseudo-randomness, ergodicity, and aperiodicity. Chaotic systems exhibit characteristics such as extreme sensitivity to initial conditions, irregular and unpredictable dynamic trajectories, and iteratively generated sequences with pseudo-random, noise-like, and aperiodic properties.[23] The application of chaotic mapping in cryptography can be traced back to 1989,[24] where the work of Matthews marked an important milestone in this field. Fridrich[25] was the first to apply chaotic systems to image cryptography in 1997, after which many researchers became interested in related technologies to maximize the application of chaotic systems in cryptography. For example, classical low-dimensional chaotic systems possess drawbacks such as short orbital periods, which reduce the complexity and unpredictability of generated pseudo-random sequences.[26] To overcome these limitations, Liu *et al.*[27] presented a spatiotemporal chaotic system with a relatively simple structure but increased parameters, complexity, and unpredictability. Moreover, Ding *et al.*[28] proposed a three-dimensional chaotic map with an adjustable Lyapunov exponent, thereby achieving a wider chaotic regime and higher system complexity. In this direction, Zhou *et al.*[29] proposed a novel n-dimensional nondegenerate discrete hyperchaotic map in which the Lyapunov exponents can be adjusted independently, providing a powerful tool for enhancing the security and robustness of encryption schemes. Recently, Wang *et al.*[30] introduced a novel chaotification approach applicable to discrete memristors, resulting in a significant enlargement of the chaotic region and enhanced dynamical complexity of discrete systems, with promising pseudo-random properties for cryptographic applications. Recent studies further suggest that not all mathematical function models suitable for cryptographic purposes possess meaningful physical interpretations of energy, and that cryptographic functions should be developed based on systems with provable energy conservation or dissipation characteristics to ensure enhanced control of attractor dynamics and system stability.[31] For example, Zhou *et al.*[32] introduced a novel hyperchaotic system and implemented it into an effective audio encryption scheme, demonstrating the feasibility of similar systems in secure communication applications. Furthermore, in addition to advances in chaotic systems, recent studies have explored hybrid image encryption models that combine chaotic systems with efficient signal processing methods, achieving improved encryption performance. For example, Liu *et al.*[33] proposed an image encryption system combining block compressive sensing with a chaotic laser system, demonstrating enhanced security and reduced computational complexity for large-scale and Internet of Things image processing applications. Despite these advances, many existing chaotic image encryption schemes still rely on low-dimensional or independently driven chaotic maps, which may suffer from limited key space, weak resistance to parameter estimation attacks, and insufficient inter-channel coupling when applied to color images. Moreover, some chaotic systems lack a clear physical basis, making their dynamic behaviors difficult to control and potentially affecting encryption stability and security. In parallel, with the continuous advancement of neural network research and the rapid development of emerging technologies such as mechanical optimization design,[34] secure synchronization control,[35] crack segmentation,[36] and brain-like functions,[37] the performance of neural networks in chaotic sequence generation has been significantly improved. Building upon the physical framework of memristive neural circuits,[38] discrete memristive neural network models with clear dynamical properties and hardware feasibility have been extensively investigated. For example, He *et al.*[39] developed a discrete memristive neural network with stable and controllable dynamics, demonstrating its effectiveness in practical intelligent applications and providing a solid foundation for secure information processing systems. Many researchers have also successfully applied the chaotic characteristics of memristive Hopfield neural networks to image encryption and related fields, achieving excellent security performance and practical application results.[40,41]

To satisfy the increasing demand for more secure and efficient encryption mechanisms, researchers have begun exploring hybrid methods that combine chaotic systems with DNA coding technology to enhance the complexity and unpredictability of encryption algorithms. DNA coding technology is based on DNA pseudo-algorithms that conceal and encrypt information, representing a new interdisciplinary branch integrating molecular biotechnology and computer technology. DNA molecules are composed of four types of nucleotides: adenine (A), thymine (T), cytosine (C), and guanine (G). Various combinations and permutations of these nucleotides can be utilized to encode information. In DNA computing, data are converted into nucleotide sequences, while operations such as addition, subtraction, and XOR are performed according to binary rules defined over nucleotide sequences to ensure the accuracy and reliability of the data.[42,43] Furthermore, integrating chaotic systems with DNA coding technology significantly

enhances the security of image encryption schemes by utilizing dynamic nonlinearity and biological encoding principles to resist cryptographic attacks.[44] Nevertheless, most existing chaos–DNA-based image encryption schemes still adopt fixed DNA coding rules or process RGB channels independently, thereby limiting encryption complexity and inter-channel diffusion capability. In addition, visually meaningful encryption schemes often face challenges in balancing visual quality and security because regular embedding patterns may expose statistical vulnerabilities. These unresolved issues motivate the development of more adaptive, strongly coupled, and visually secure medical image encryption mechanisms. Therefore, to address the limitations of existing color image encryption schemes, this paper proposes a visually meaningful medical image encryption scheme based on image steganography and memristive Hopfield neural networks.

The main contributions of this paper can be summarized as follows:

(i) A multi-channel color image pixel permutation algorithm driven by chaotic sequences is developed. Unlike conventional schemes that process RGB channels independently, the proposed algorithm adopts a dynamically key-controlled pseudo-random permutation strategy to perform joint cross-channel pixel permutation among RGB channels, thereby enhancing inter-channel coupling and confusion capability.

(ii) A double-layer diffusion mechanism integrating cellular automata and dynamic DNA coding is constructed. Instead of treating these techniques as independent diffusion modules, chaotic sequences are employed to dynamically control the evolution of cellular automaton states for generating pseudo-random diffusion matrices, which are then adaptively combined with dynamic DNA coding rules to diffuse pixel values.

(iii) A dynamic multi-channel pseudo-random least significant bit (LSB) embedding algorithm based on chaotic sequences is proposed. Unlike traditional LSB substitution methods, the proposed algorithm employs a cross-channel embedding and exchange strategy that enables pseudo-random interaction of cipher information among different color channels, such as RGB channels.

(iv) Comprehensive performance evaluations and statistical analyses demonstrate that the proposed encryption algorithm achieves high security performance and strong resistance against various cryptanalytic attacks.

The rest of this paper is organized as follows: Section 2 describes the preliminary work of the encryption algorithm. Section 3 details the encryption procedure. Section 4 presents numerical experiments to verify the security and feasibility of the proposed encryption algorithm. Finally, Section 5 summarizes the conclusions of this work.

2. Preliminary works

Figure 1 presents the flow framework of the proposed encryption algorithm. The proposed encryption algorithm can be roughly divided into two parts. One part generates the chaotic sequence S using the key K, while the other performs image encryption based on the algorithm and the original image. In this section, the methods for generating the key K and chaotic sequence S are described.

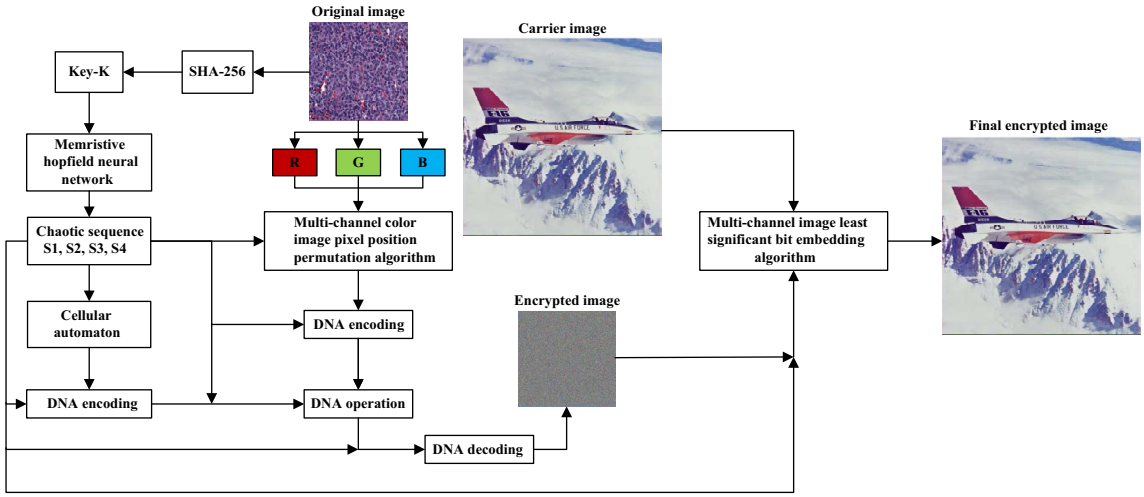


Fig. 1. Flowchart of the proposed algorithm.

2.1. Secure hash algorithm

A hash function is a type of function that maps input data to a fixed-length hash value. Due to the avalanche effect, even a slight change in the input data will result in a completely different hash value. In image encryption applications, the image is used as the input of the hash function, and the generated hash value serves as the encryption key for the image. By utilizing this property of hash functions, a unique key corresponding to a specific image can be generated, thereby enabling an encryption scheme based on the principle of "one image, one key".

To ensure the secure transmission and storage of the image-specific key, the following approaches are adopted. For secure key transmission, the key is encrypted using an asymmetric encryption algorithm. The encrypted key is then transmitted together with the encrypted image, and the receiver uses the corresponding private key to decrypt the key. For secure storage, the key is stored using a trusted platform module (TPM). Specifically, the key can either be securely stored inside the TPM or encrypted and stored in a secure database.

Among the various hash algorithms, the SHA-2 family possesses high security, flexibility, efficiency, and strong collision resistance. Therefore, it has been widely applied in cryptography and information security. In this paper, the SHA-256 function from the SHA-2 family is first employed to generate the initial key of the image encryption algorithm. The pseudo-code for the key generation process is presented in Algorithm 1.

Algorithm 1 Obtain key process.

Input: Plain image I.
Output: Initial key K.

$H_1 \leftarrow \text{SHA-256}(I)$;
for $(i = 1{:}8{:}256)$ **do**
  $H_2^{(k+7)/8} \leftarrow \text{bin2dec}(H_1(i : i+7))$;
end for
for $(i = 1{:}4{:}32)$ **do**
  $k_{(i+3)/4} \leftarrow \frac{H_2^k + H_2^{k+1} + H_2^{k+2} + H_2^{k+3}}{256}$;
end for
$K = [k_1, k_2, \ldots, k_8]$;

The algorithm described above utilizes the SHA-256 hash function to process the plaintext image and generate a 256-bit hash value. Subsequently, the 256-bit hash value is divided into 32 groups, and each group is converted into its corresponding decimal value. These 32 decimal values are then aggregated in groups of four, and each sum is divided by 256 to obtain the keys k_i. Finally, the initial key K is constructed by combining k_1–k_8.

2.2. Memristive hopfield neural network

The Hopfield neural network (HNN), introduced by Hopfield[45] in 1982, is an artificial neural network model that describes the coupling behavior among neurons. Subsequent studies incorporated advances in neuromorphic computing by integrating memristors, which possess nonlinear circuit characteristics and self-memory effects, into the synaptic connectivity matrices of HNNs. This development led to the emergence of memristive Hopfield neural networks (MHNNs). Owing to their inherent multistability, nonlinear dynamical behavior, and strong adjustability, MHNNs are capable of generating chaotic sequences with higher randomness and stronger instability, thereby enriching and diversifying chaotic dynamics. Therefore, MHNNs have been widely utilized in image encryption applications for generating complex chaotic behavior. In this paper, the MHNN model proposed in Ref. [16] is adopted. Compared with conventional chaotic maps, this system exhibits superior dynamical characteristics, making it more suitable for image encryption applications. The system demonstrates hyperchaotic behavior with multiple positive Lyapunov exponents, thereby ensuring higher complexity and greater sensitivity than low-dimensional chaotic systems. Furthermore, the introduction of memristive autapses enables the system to generate uncountably infinite coexisting attractors determined by the initial internal parameters (z_1, z_2). The MHNN model used for chaotic sequence generation is expressed as follows:

$$\begin{cases} \dot{x}_1 = -x_1 + \tanh(x_1) + 0.5\tanh(x_2) - 2\tanh(x_3) \\ \qquad - \tanh(x_4), \\ \dot{x}_2 = -x_2 + 2.3\tanh(x_2) + 3\tanh(x_3), \\ \dot{x}_3 = -0.7x_3 + \rho_2 G_2 \tanh(x_1) - 3\tanh(x_2) \\ \qquad + \tanh(x_3) - \tanh(x_4), \\ \dot{x}_4 = -x_4 + \rho_1 G_1 \tanh(x_1) + 3\tanh(x_2) \\ \qquad + \tanh(x_3) + 0.3\tanh(x_4), \\ \dot{z}_1 = 3\cos(z_1)\tanh(z_1) - \tanh(x_1), \\ \dot{z}_2 = 3\cos(z_2)\tanh(z_2) - \tanh(x_1), \end{cases} \tag{1}$$

where x_1–x_4 denote the membrane potentials of neurons N_1–N_4, reflecting the dynamic behaviors of the neurons. The variables z_1 and z_2 represent the internal states of the memristive synapses and are used to control the conductances G_1 and G_2. The introduction of the memristor-based state variables z_1 and z_2 is essential for achieving adaptive synaptic coupling, thereby enabling infinite equilibrium points and hyperchaotic behavior. The expressions for G_1, G_2, ρ_1, and ρ_2 are given as follows:

$$\begin{cases} G_1 = 500z_1 + 0.4\cos(z_1), \\ G_2 = 500z_2 + 0.4\cos(z_2), \\ \rho_1 = \mod(k_1 + k_3 + k_5 + k_7, 2.6) + 0.003, \\ \rho_2 = \mod(k_2 + k_4 + k_6 + k_8, 0.064) + 0.001. \end{cases} \tag{2}$$

To demonstrate the suitability of the MHNN for cryptographic applications, the dynamical properties of the system with respect to the parameter ρ_1 are analyzed in this paper. Figure 2(a) presents the bifurcation diagram of the state variable x_2. It can be observed that when $\rho_1 > 0$, the system enters a dense chaotic state, resulting in complex non-periodic behavior that is suitable for random sequence generation. This phenomenon is further confirmed by the Lyapunov exponent (LE) spectrum shown in Fig. 2(b). Specifically, when $\rho_1 > 0$, the largest Lyapunov exponent (LE_1, represented by the yellow curve) remains positive, confirming the chaotic nature of the system. These results indicate that the system exhibits hyperchaotic behavior, which provides higher sequence complexity than conventional chaotic systems.

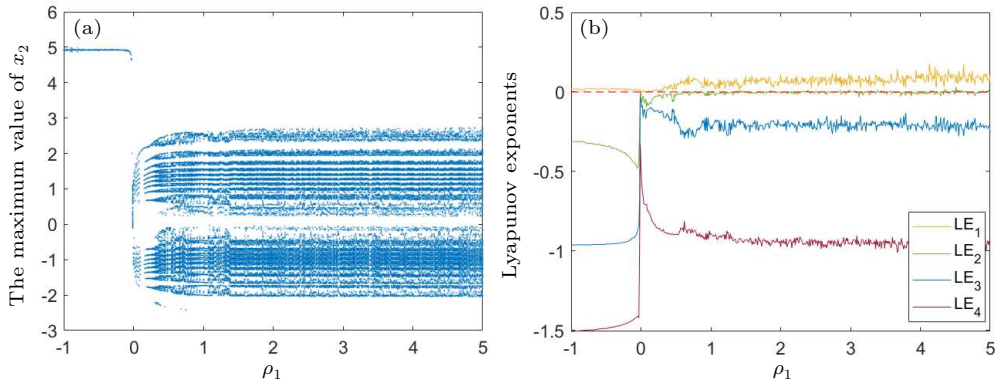


Fig. 2. The dynamics of Lyapunov exponents and the maximum value of x_2 with respect to ρ_1.

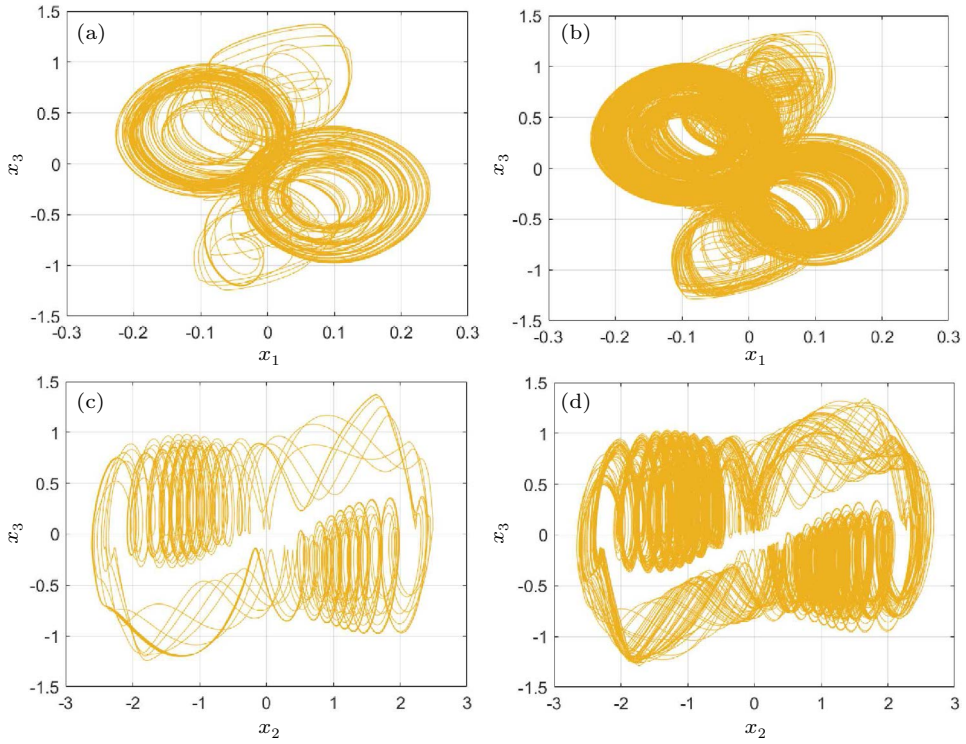


Fig. 3. The 2D phase portraits projected onto the x_1–x_3 and x_2–x_3 planes.

To further verify the suitability of the MHNN for encryption and random sequence generation, the phase-space trajectories of the system are analyzed. Figure 3 illustrates the two-dimensional phase portraits of the MHNN under different parameter settings determined by the key generation mechanism. Specifically, in Figs. 3(a) and 3(c), the parameters are set to $\rho_1 = 1.2$ and $\rho_2 = 0.05$, while in Figs. 3(b) and 3(d), the parameters are set to $\rho_1 = 1.9$ and $\rho_2 = 0.05$. In both cases, the system exhibits strong ergodicity, indicating robust chaotic behavior characterized by dense, bounded, and aperiodic phase-space trajectories. These trajectories densely cover the phase space, thereby enabling the generation of highly random sequences for effective image pixel scrambling.

Using the MHNN model described above, the key K is employed as the system parameter input, and the resulting outputs $\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$, and $\dot{x}_4$ are utilized as chaotic sequences S_1, S_2, S_3, and S_4, respectively.

3. Encryption scheme

In this section, the permutation, diffusion, and steganography algorithms used for color medical image encryption are introduced. It should be noted that the image decryption process is essentially the inverse of the encryption operation.

3.1. Image permutation process

Random pixel permutation is an image encryption technique used to enhance image security by introducing randomness into pixel positions, thereby concealing the spatial information of the image. Traditional methods usually perform repetitive permutation operations independently on the three channels of a color image using key-dependent sorting sequences, resulting in limited correlation with the image content and relatively low efficiency. In contrast, the proposed method designs a multi-channel color image pixel permutation

algorithm based on chaotic-sequence-driven pseudo-random permutation and dynamically generated keys. The proposed algorithm effectively conceals the spatial relationships among image pixels, thereby improving the security of the encryption process. The pseudo-code for the image permutation process is presented in Algorithm 2.

Algorithm 2 Permutation process.

Input: Plain image I, Initial key K, Chaotic sequence S_1, S_2, S_3.

Output: Permuted image I_1.

$[m,n] \leftarrow \text{size}(I)$;
$[R_1,G_1,B_1] \leftarrow \text{imsplit}(I)$;
for $(k = 3001{:}1{:}m \times n + 3000)$ **do**
    $P_1 \leftarrow \text{mod}\,(\text{floor}(\text{abs}(S_1^k) \times 10^7),\ 3) + 1$;
    $P_2 \leftarrow \text{mod}\,(\text{floor}(\text{abs}(S_2^k) \times 10^7),\ 3) + 1$;
    $P_3 \leftarrow \text{mod}\,(\text{floor}(\text{abs}(S_3^k) \times 10^7),\ 3) + 1$;
end for
$\text{key}_1 \leftarrow \text{round}(\text{abs}(k_1 \times k_2 \times k_3))$;
$\text{key}_2 \leftarrow \text{round}(\text{abs}(k_4 \times k_5 \times k_6))$;
$\text{key}_3 \leftarrow \text{round}(\text{abs}(k_7 \times k_8))$;
seed $\leftarrow \text{mod}\,(key_1)$;
rand_index $\leftarrow \text{reshape}(\text{randperm}(m \times n), m, n)$;
for $(i = 1{:}1{:}m)$ **do**
    for $(j = 1{:}1{:}n)$ **do**
        $(h,g) \leftarrow (\text{floor}((\text{rand_index}(i,j)-1)/n)+1, \text{mod}\,(\text{rand_index}(i,j)-1,n)+1)$;
    end for
end for
if $(P_1(i,j) == 1)$ **then**
    $R_1(i,j) \leftrightarrow R_1(h,g)$;
else if $(P_1(i,j) == 2)$ **then**
    $R_1(i,j) \leftrightarrow G_1(h,g)$;
else if $(P_1(i,j) == 3)$ **then**
    $R_1(i,j) \leftrightarrow B_1(h,g)$;
end if
seed $\leftarrow \text{mod}(\text{key}_2)$;
rand_index $\leftarrow \text{reshape}(\text{randperm}(m \times n), m, n)$;
for $(i = 1{:}1{:}m)$ **do**
    for $(j = 1{:}1{:}n)$ **do**
        $(h,g) \leftarrow (\text{floor}((\text{rand_index}(i,j)-1)/n)+1, \text{mod}\,(\text{rand_index}(i,j)-1,n)+1)$;
    end for
end for
if $(P_2(i,j) == 1)$ **then**
    $G_1(i,j) \leftrightarrow G_1(h,g)$;
else if $(P_2(i,j) == 2)$ **then**
    $G_1(i,j) \leftrightarrow B_1(h,g)$;
else if $(P_2(i,j) == 3)$ **then**
    $G_1(i,j) \leftrightarrow R_1(h,g)$;
end if
seed $\leftarrow \text{mod}(\text{key}_3)$;
rand_index $\leftarrow \text{reshape}(\text{randperm}(m \times n), m, n)$;

Algorithm 2 Continued.

for $(i = 1{:}1{:}m)$ **do**
    for $(j = 1{:}1{:}n)$ **do**
        $(h,g) \leftarrow (\text{floor}((\text{rand_index}(i,j)-1)/n)+1, \text{mod}\,(\text{rand_index}(i,j)-1,n)+1)$;
    end for
end for
if $(P_3(i,j) == 1)$ **then**
    $B_1(i,j) \leftrightarrow B_1(h,g)$;
else if $(P_3(i,j) == 2)$ **then**
    $B_1(i,j) \leftrightarrow R_1(h,g)$;
else if $(P_3(i,j) == 3)$ **then**
    $B_1(i,j) \leftrightarrow G_1(h,g)$;
end if
$I_1 \leftarrow \text{cat}[R_1,G_1,B_1]$;

In the proposed algorithm, to ensure precise control of channel switching between pixels, a subsequence with a length of $m \times n$ is extracted starting from the 3000th iteration. This strategy is adopted to eliminate the transient effects of the chaotic system during the initial stage, thereby ensuring that the obtained subsequence possesses desirable randomness and stability. Based on this subsequence, the `randperm` function is employed to generate a non-repetitive random integer permutation sequence. By performing integer division and modulo operations on each element of the sequence with respect to a positive integer n, the sequence can be transformed into a set of two-dimensional coordinates. Owing to the characteristics of the `randperm` function, all coordinates within the space can be visited randomly and uniquely in a single round. The key K is used as the seed for generating the pseudo-random sequence through the `randperm` function. This strategy not only guarantees the reproducibility of pseudo-random sequence generation but also ensures that a unique pseudo-random sequence is generated for each image, thereby enhancing the security and uniqueness of the encryption system.

3.2. Image diffusion process

Before introducing the pixel diffusion mechanism designed for color images, a brief overview of the underlying cellular automaton algorithm is first provided to clarify its fundamental operating principles.

3.2.1. Cellular automaton

Cellular automaton (CA) is a discrete dynamical system model composed of cells, neighborhoods, and state transition rules. It operates in discrete time and discrete space, where the evolution of each cell is determined according to a set of fixed rules and interactions with neighboring cells. Despite its simple structure, CA can generate highly complex dynamical behaviors. Owing to this property, CA has been widely applied in image encryption schemes. In this paper, a two-dimensional Moore-type cellular automaton is adopted, in which cells are arranged in a grid structure and interact with their eight nearest

neighbors. This property is particularly important for image encryption because it significantly enhances the randomness and security of the encryption scheme. The pseudo-code for the cellular automaton process is presented in Algorithm 3.

Algorithm 3 Cellular automaton process.

Input: Chaotic sequence S_4.
Output: Cell matrix C_1.

for ($k = 3001{:}1{:}m \times n + 3000$) **do**
  $O_1 \leftarrow \text{mod}(\text{floor}(\text{abs}(S_4^k) \times 1000), 2)$;
end for
$O_1 \leftarrow \text{reshape}(O_1, m, n)$;
for ($i = 1{:}1{:}m$) **do**
  for ($j = 1{:}1{:}n$) **do**
    $N_1 \leftarrow \text{neighbor}(O_1(i, j))$;
    for ($a = 1{:}1{:}3$) **do**
      for ($b = 1{:}1{:}3$) **do**
        if ($a \neq 2$ and $b \neq 2$) **then**
          $N_1(a, b) \leftarrow \text{transition}(N_1)$;
          $N_2 \leftarrow \text{strcat}(N_2, \text{num2str}(N_1(a, b)))$;
        end if
      end for
    end for
    $O_1 \leftarrow \text{restore}(N_1)$;
    $C_1(i, j) \leftarrow \text{bin2dec}(N_2)$;
  end for
end for

In this algorithm, the cell set P_1 is generated using the chaotic sequence S_4, where the state value of each cell is restricted to either 0 or 1. Subsequently, the neighbor function is applied to obtain the values of cells within a 3×3 neighborhood matrix surrounding the target cell. When the target cell is located near the matrix boundary, the corresponding values from the opposite boundary are employed to complete the neighborhood range. For example, when the target cell is located in the first column of the matrix, the neighboring cell values from the last column in the corresponding row positions are utilized for padding, thereby enabling the construction of a complete 3×3 neighborhood matrix.

Afterward, a transition function is employed to update the states of the eight neighboring cells within the 3×3 neighborhood matrix while excluding the central cell. The update process traverses the neighboring cells in a circular order, where the XOR operation of the subsequent four neighboring cells is performed to determine the updated state of the current cell. After processing all eight neighboring cells, a new 3×3 cell matrix N_1 is obtained. The `restore` function is then applied to write the elements of N_1 back into the corresponding positions of P_1, thereby completing the state update of the cell set. Finally, the eight neighboring elements surrounding the center cell in N_1 are extracted and serialized to form an 8-bit binary string N_2. The decimal representation of N_2 is subsequently used to construct the matrix C_1, which serves as the mapping matrix for the subsequent diffusion process.

3.2.2. Diffusion algorithm

During the diffusion phase of image encryption, this paper proposes a novel algorithm that combines DNA coding technology with cellular automata. DNA coding technology utilizes the complementary base-pairing characteristics of DNA molecules to encode image pixel values into DNA sequences. Information transformation is then performed according to DNA operation rules and a mapping matrix, after which the transformed data are decoded to restore the image information, thereby achieving image diffusion. This mechanism not only enhances the complexity of the encryption system but also improves its resistance against potential attacks. The pseudocode for the diffusion process is presented in Algorithm 4.

Algorithm 4 Diffusion process.

Input: Permuted image I_1, Cell matrix C_1, Chaotic sequence S_1, S_2, S_3.
Output: Diffusion image I_2.

$[R_2, G_2, B_2] \leftarrow \text{imsplit}(I_1)$;
for ($k = 3001{:}1{:}m \times n + 3000$) **do**
  $D_1 \leftarrow \text{mod}(\text{floor}(\text{abs}(S_1^k) \times 1000), 8) + 1$;
  $D_2 \leftarrow \text{mod}(\text{floor}(\text{abs}(S_2^k) \times 1000), 8) + 1$;
  $D_3 \leftarrow \text{mod}(\text{floor}(\text{abs}(S_3^k) \times 1000), 4) + 1$;
end for
$D_1 \leftarrow \text{reshape}(D_1, m, n)$;
$D_2 \leftarrow \text{reshape}(D_2, m, n)$;
$D_3 \leftarrow \text{reshape}(D_3, m, n)$;
$E \leftarrow \text{dec2bin}(C_1)$;
Introducing DNA encoding and decoding rule T1;
Introducing DNA operation rule T2;
$R_2 \leftarrow \text{dec2bin}(R_2)$;
for ($i = 1{:}1{:}m$) **do**
  for ($j = 1{:}1{:}n$) **do**
    $\text{DNA}_1 \leftarrow T_1(D_1(i, j), R_2(i, j))$;
    $\text{DNA}_2 \leftarrow T_1(D_1(i, j), E(i, j))$;
    $\text{DNA}_3 \leftarrow T_2(D_3(i, j), \text{DNA}_1, \text{DNA}_2)$;
    $R_2(i, j) \leftarrow T_1(D_2(i, j), \text{DNA}_3)$;
  end for
end for
$R_2 \leftarrow \text{bin2dec}(R_2)$;
Replace R_2 with G_2 or B_2, repeat step 13 to 22 to get new G_2, B_2;
$I_2 \leftarrow \text{cat}[R_2, G_2, B_2]$;

Table 1. DNA coding rules.

Binary \ Rule	1	2	3	4	5	6	7	8
00	A	A	T	T	C	G	C	G
01	C	G	C	G	A	A	T	T
10	G	C	G	C	T	T	A	A
11	T	T	A	A	G	C	G	C

In the proposed algorithm, three chaotic sequences, namely S_1, S_2, and S_3, are employed to generate the corresponding driving sequences D_1, D_2, and D_3. These driving sequences dynamically control the DNA encoding, decoding,

and operational procedures, thereby significantly improving the complexity and security of the algorithm. The DNA coding rules are listed in Table 1, which contains eight encoding cases. Table 2 presents the corresponding DNA operation rules under Rule 1. Furthermore, the target image is converted into two DNA sequences, namely DNA_1 and DNA_2, under the control of the driving sequence D_1 according to the DNA encoding rule T_1. Subsequently, driven by the sequence D_3, DNA_1 and DNA_2 are processed according to the DNA operation rule T_2 to generate a new DNA sequence, denoted as DNA_3. Finally, under the control of the driving sequence D_2, DNA_3 is decoded according to the encoding rule T_1 to obtain the final diffused image. This process significantly enhances the nonlinearity and security of image encryption.

Table 2. The corresponding DNA operation rules in the case of rule 1.

Operation \ Base	(A,A)	(A,T)	(A,C)	(A,G)	(T,A)	(T,T)	(T,C)	(T,G)	(C,A)	(C,T)	(C,C)	(C,G)	(G,A)	(G,T)	(G,C)	(G,G)
$+$	A	T	C	G	T	G	C	A	C	A	G	T	G	C	T	A
$-$	A	T	C	G	C	A	G	T	T	G	C	A	G	C	T	A
$\oplus$	A	T	C	G	T	A	G	C	C	G	A	T	G	C	T	A
$\odot$	T	A	G	C	A	T	C	G	G	C	T	A	C	G	A	T

3.3. Image steganography process

Image steganography can be employed to embed encrypted images into carrier images, thereby generating visually meaningful images instead of images exhibiting obvious noise-like patterns. This process effectively conceals encrypted information and reduces the probability of detection. One of the most commonly used image steganography methods is LSB embedding, in which secret information is hidden within the least significant bits of image pixels. This technique offers advantages such as high imperceptibility, ease of implementation, and robustness against certain image processing operations. By replacing the least significant bits of the original image with secret information, the visual quality and natural appearance of the image can be preserved while invisibly embedding confidential data.

In this paper, a novel dynamic multi-channel pseudo-random LSB embedding method is proposed. The proposed approach adopts a cross-channel embedding strategy in which chaotic sequences are utilized to drive pseudo-random channel exchange operations. This mechanism effectively reduces repetitive embedding patterns that may occur during the embedding process, thereby preserving the visual naturalness and originality of the image. The pseudo-code for the least significant bit embedding process is presented in Algorithm 5.

In the proposed algorithm, the driving sequence L_1 is generated from the chaotic sequence S_4. To facilitate a clearer understanding of the proposed method, the following points are clarified. First, to ensure that all information from the target image can be completely embedded, the width and height of the carrier image Q are required to satisfy $M = 2m$ and $N = 2n$, respectively, where M and N denote the width and height of the carrier image, while m and n denote those of the target image.

Second, during the LSB substitution process, the sequences F_2, F_6, and F_{10} are utilized for processing H_2. Similarly, for H_3 and H_4, the corresponding sequences are F_3, F_7, and F_{11}, as well as F_4, F_8, and F_{12}, respectively. Finally, when processing G_4 and B_4, the mapping relationship corresponding to L_1 must be appropriately transformed to ensure that all in-

Algorithm 5 Least significant bit process.

Input: Diffusion image I_2, Carrier image Q, Chaotic sequence S_4.

Output: Embedded image I_3.

$[M,N] \leftarrow \mathrm{size}(Q)$;
$[R_3,G_3,B_3] \leftarrow \mathrm{imsplit}(I_2)$;
$[R_3,G_3,B_3] \leftarrow \mathrm{dec2bin}([R_3,G_3,B_3])$;
$[R_4,G_4,B_4] \leftarrow \mathrm{imsplit}(Q)$;
$[R_4,G_4,B_4] \leftarrow \mathrm{dec2bin}([R_4,G_4,B_4])$;
for $(k = 3001{:}1{:}m \times n + 3000)$ **do**
    $L_1 \leftarrow \mathrm{mod}(\mathrm{floor}(\mathrm{abs}(S_4^k) \times 10^7),\ 3) + 1$;
end for
Split the eight-bit binary number from R_3 equally into four two-bit binary arrays, resulting in F_1–F_4.
Repeat the above operation on G_3 and B_3 to obtain F_5–F_8, F_9–F_{12}.
$F_1 \leftarrow R_3(1:2); F_5 \leftarrow G_3(1:2); F_9 \leftarrow B_3(1:2)$;
$F_2 \leftarrow R_3(3:4); F_6 \leftarrow G_3(3:4); F_{10} \leftarrow B_3(3:4)$;
$F_3 \leftarrow R_3(5:6); F_7 \leftarrow G_3(5:6); F_{11} \leftarrow B_3(5:6)$;
$F_4 \leftarrow R_3(7:8); F_8 \leftarrow G_3(7:8); F_{12} \leftarrow B_3(7:8)$;
Decompose R_4 into four two-dimensional binary arrays H_1, H_2, H_3, H_4 of size $m \times n$;
Replace the last two binary digits of H_1 with those from F_1 or F_5 or F_9;
for $(i = 1{:}1{:}m \times n)$ **do**
    if $(L_1(i) = 1)$ **then**
        $H_1(7:8) \leftarrow F_1$;
    else if $(L_1(i) = 2)$ **then**
        $H_1(7:8) \leftarrow F_5$;
    else$(L_1(i) = 3)$
        $H_1(7:8) \leftarrow F_9$;
    end if
end for
Repeat step 17 to 25 to obtain new H_2, H_3, and H_4;
$R_4 \leftarrow \mathrm{strcat}([H_1,H_2,H_3,H_4])$;
$R_4 \leftarrow \mathrm{bin2dec}(R_4)$;
Repeat step 15 to 28 to obtain new G_4 and B_4;
$I_3 \leftarrow \mathrm{cat}([R_4,G_4,B_4])$;

formation from the target image can be completely concealed. The specific mapping rules are listed in Table 3. This transformation strategy further enhances the flexibility and security of the proposed algorithm.

Table 3. L_1 mapping rules.

	R_4	G_4	B_4
$L_1 = 1$	F_1–F_4	F_5–F_8	F_9-F_{12}
$L_1 = 2$	F_5–F_8	F_9–F_{12}	F_1-F_4
$L_1 = 3$	F_9–F_{12}	F_1–F_4	F_5-F_8

4. Experiment result and analysis

4.1. Experimental environment

The experiments in this paper were conducted on a laptop computer running Windows 11, equipped with a 13th Gen Intel® Core™ i9-13900HX processor, 16 GB RAM, and a 954 GB solid-state drive (SSD). The proposed algorithm was implemented and tested using MATLAB R2016a. In the experiments, the tested color medical images were selected from the Eye Disease Deep Learning Dataset[46] and the Kumar Tumor Dataset.[47] Four RGB images were used for the experiments, including Eye disease (size $512 \times 512 \times 3$), Pepper (size $512 \times 512 \times 3$), Baboon (size $512 \times 512 \times 3$), and Kumar tumor (size $768 \times 768 \times 3$).

4.2. Simulation results

The secret key was initialized as K. Figures 4(a)–4(d) present the original images of Eye disease, Pepper, Baboon, and Kumar tumor. Figures 4(e)–4(h) show the corresponding encrypted images. It can be clearly observed that all encrypted images exhibit noise-like characteristics, are visually unrecognizable, and reveal no meaningful image features. Figures 4(i)–4(l) present the corresponding decrypted images.

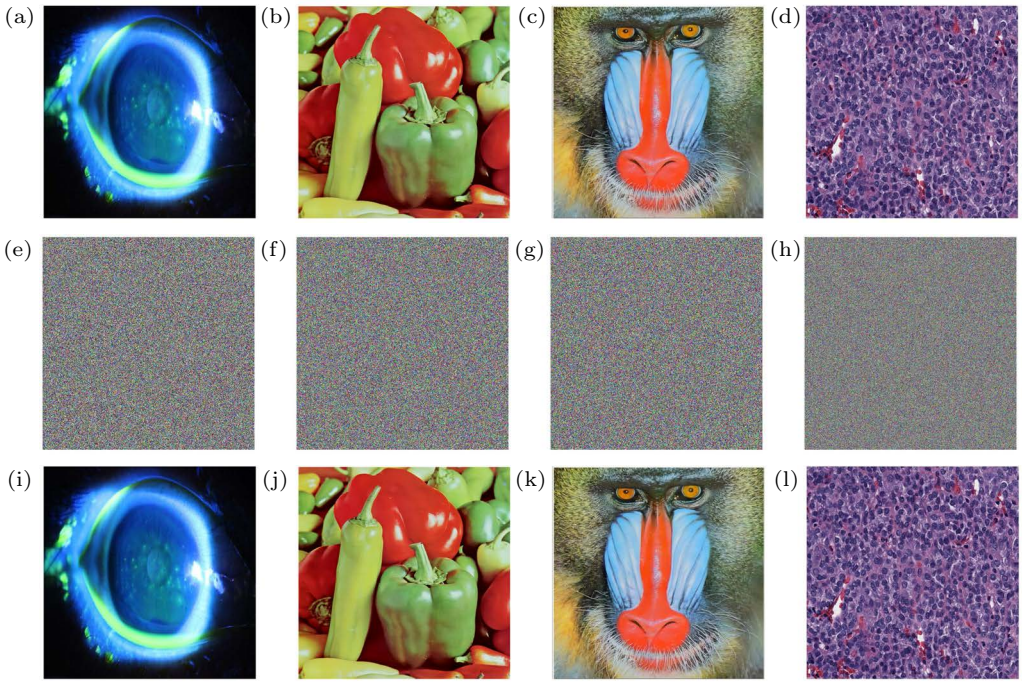


Fig. 4. Encryption and Decryption Process test. (a)–(d) Original images; (e)–(h) Encrypted images; (i)–(l) Decrypted images.

4.3. Key space and sensitivity analysis

The key space refers to the set of all possible keys in an encryption system. In cryptography, a sufficiently large key space is essential for resisting brute-force attacks. Generally, a secure encryption algorithm should possess a key space no smaller than 2^{100},[48] thereby providing effective protection against exhaustive brute-force attacks. In this paper, the key consists of a 256-bit hash value and four system parameters, namely G_1, G_2, ρ_1, and ρ_2. Each system parameter is defined over a continuous interval with a computational precision of 10^{-9}. Under this precision, each parameter can take approximately 10^9 distinct effective values, resulting in a key space of 10^9 for each parameter. Therefore, the total key space of the proposed encryption algorithm is calculated as

$$2^{256} \times 10^{9\times 4} \approx 2^{376},$$

which is significantly larger than 2^{100}. Consequently, the proposed encryption algorithm possesses a sufficiently large key space to effectively resist brute-force attacks.

In secure encryption systems, key sensitivity is another critical property. For a highly secure cryptosystem, even a slight variation in the key should produce encryption and decryption results that differ significantly from those generated using the original key. Taking the Baboon image as an example, when the initial key parameters G_1, G_2, ρ_1, and ρ_2 are perturbed within the computational precision of 10^{-9}, respec-

tively, the image cannot be correctly decrypted, as illustrated in Fig. 5. These results demonstrate that the proposed algorithm exhibits high key sensitivity and can effectively resist brute-force attacks.

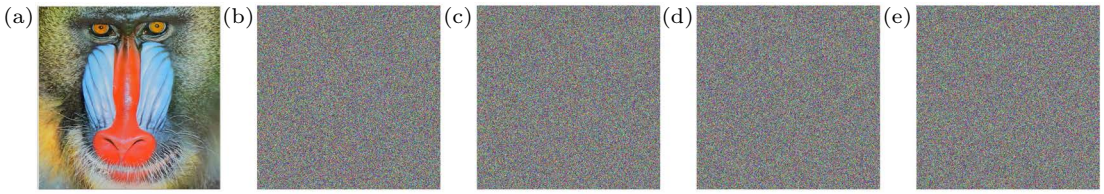


Fig. 5. Key sensitivity test. (a) Correctly decrypted image; (b) Decrypted image with $G_1 + 1 \times 10^{-9}$; (c) Decrypted image with $G_2 + 1 \times 10^{-9}$; (d) Decrypted image with $\rho_1 + 1 \times 10^{-9}$; (e) Decrypted image with $\rho_2 + 1 \times 10^{-9}$.

4.4. Histogram analysis

The histogram of an image is an important statistical characteristic that represents the frequency distribution of pixel values. Generally, the more uniform the histogram distribution is, the less statistical information about the plaintext image can be inferred by attackers. Figures 6(a)–6(d) show the histograms of the plaintext images Eye disease, Pepper, Baboon, and Kumar tumor for the R, G, and B color channels. Figures 6(e)–6(h) present the corresponding histograms of the encrypted images for the R, G, and B color channels. As shown in Fig. 6, the histogram distributions of the R, G, and B color channels in the plaintext images are uneven, with noticeable gaps among pixel values. In contrast, the histogram distributions of the encrypted images are relatively uniform, and the pixel values are distributed more evenly. These results indicate that the proposed algorithm effectively conceals the statistical characteristics of image pixels and demonstrates strong resistance against statistical analysis attacks.

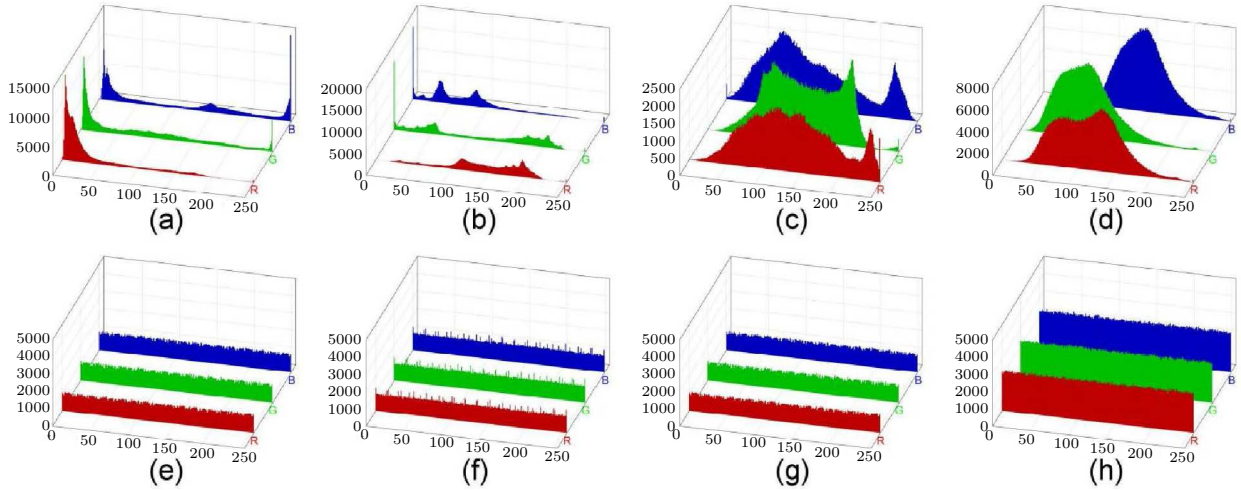


Fig. 6. Histogram analysis test. (a)–(d) The histograms of plaintext images: Eye disease, Pepper, Baboon, and Kumar tumor, respectively. (e)–(h) The histograms corresponding to the ciphertext images.

4.5. Pixel correlation analysis

Pixel correlation refers to the similarity between adjacent pixels in an image. In general, adjacent pixels in a plaintext image exhibit strong correlation. Therefore, to enhance the security of cipher images, encryption algorithms are expected to effectively destroy the strong correlation among adjacent pixels. Moreover, the closer the correlation coefficient of adjacent pixels in the encrypted image is to zero, the more difficult it is for attackers to extract information about the original image. This results in improved security and stronger encryption performance.

To compare the correlation between adjacent pixels before and after encryption, 10,000 pairs of adjacent pixels are randomly selected in this paper. The correlation coefficients in the horizontal, vertical, and diagonal directions are then calculated for both plaintext and ciphertext images. The correlation coefficient is defined as follows:

$$\begin{cases} r_{ab} = \dfrac{\mathrm{cov}(a,b)}{\sqrt{D(a)}\sqrt{D(b)}}, \\ \mathrm{cov}(a,b) = \frac{1}{N}\sum_{i=1}^{N}(a_i - E(a))(b_i - E(b)), \\ D(a) = \frac{1}{N}\sum_{i=1}^{N}(a_i - E(a))^2, \\ E(a) = \frac{1}{N}\sum_{i=1}^{N} a_i, \end{cases} \tag{3}$$

where a and b represent the grayscale values of two adjacent pixels. $E(a)$ and $D(a)$ denote the mathematical expectation and variance of a, respectively. $\mathrm{Cov}(a,b)$ denotes the covariance between a and b. Figure 7 illustrates the distribution of adjacent pixels in the original images (Eye disease, Pepper, Baboon, and Kumar tumor) and their corresponding encrypted images. It is evident that the distribution of adjacent pixels in

plaintext images is highly concentrated along a straight line, indicating strong correlation. In contrast, the distribution of adjacent pixels in encrypted images appears randomly scattered, indicating a significant reduction in correlation.

Table 4 also presents the detailed statistical results. It can be observed that plaintext images exhibit high correlation coefficients in the horizontal, vertical, and diagonal directions across all RGB channels, with values close to 1. In contrast, the corresponding correlation coefficients of the encrypted images are close to 0, indicating a substantial reduction in pixel correlation. These results demonstrate that the proposed encryption scheme effectively destroys the correlation among adjacent pixels in the original images and provides strong resistance against statistical attacks.

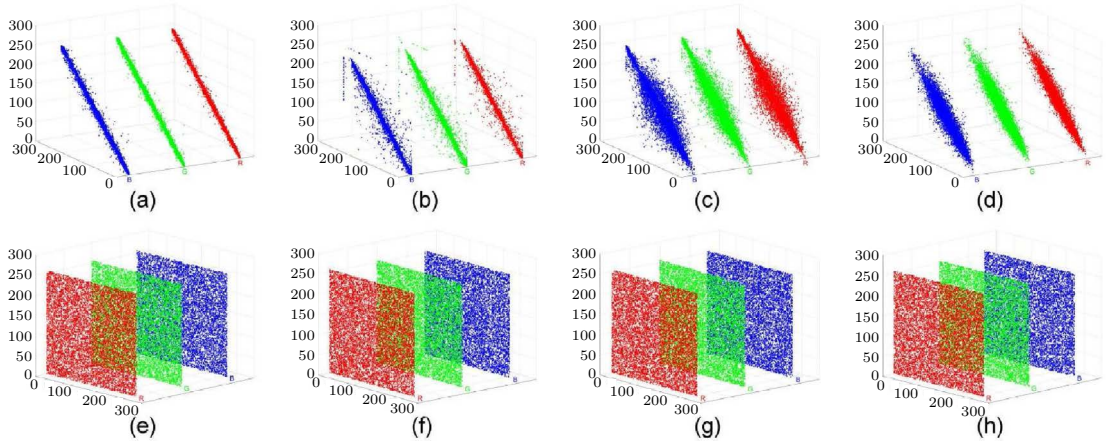


Fig. 7. Pixel correlation analysis test. (a)–(d) The correlation distributions of plaintext images Eye disease, Pepper, Baboon, and Kumar tumor, respectively. (e)–(h) The corresponding correlation distributions of the ciphertext images.

Table 4. The results of correlation analysis of Baboon image.

Image		Original image			Encrypted image		
		R	G	B	R	G	B
Eye disease	Horizontal	0.997177	0.998847	0.998693	−0.001985	−0.002465	−0.000095
	Vertical	0.996314	0.998344	0.998626	−0.001240	−0.000552	0.000145
	Diagonal	0.994215	0.997328	0.997593	−0.000840	−0.000380	0.002795
Pepper	Horizontal	0.985433	0.989475	0.979427	−0.000063	0.000620	0.007157
	Vertical	0.983527	0.985579	0.973569	−0.001012	0.002809	0.000298
	Diagonal	0.972208	0.977686	0.953453	−0.002239	−0.002355	0.002723
Baboon	Horizontal	0.897780	0.846742	0.908830	−0.002111	−0.000122	0.000452
	Vertical	0.937716	0.908670	0.941133	−0.008649	−0.002025	0.003240
	Diagonal	0.873070	0.814266	0.883681	0.000468	0.008631	−0.003715
Kumar tumor	Horizontal	0.950059	0.939657	0.928132	0.000291	0.001160	0.001961
	Vertical	0.929798	0.918054	0.902228	0.000437	−0.001204	0.000389
	Diagonal	0.889079	0.874506	0.845184	−0.003142	−0.000692	0.000146

4.6. Information entropy analysis

Information entropy is widely used to measure the uncertainty inherent in data. In general, higher uncertainty corresponds to higher information entropy. In image encryption, information entropy is an important metric for evaluating the randomness of encrypted images. The theoretical maximum value of information entropy is 8. Typically, if the pixel distribution of an image is more uniform, the information entropy will be higher and closer to the ideal value of 8. This indicates greater complexity in the image, making it more difficult for a potential attacker to decrypt it. The formula for information entropy is given as follows:

$$H(s) = \sum_{i=1}^{256} P(s_i) \log_2 \frac{1}{P(s_i)}, \tag{4}$$

Table 5. Information entropy of test images.

Image	Original image			Encrypted image		
	R	G	B	R	G	B
Eye	6.3239	7.3527	7.3036	7.9971	7.9971	7.9973
Pepper	7.3486	7.4895	7.0356	7.9929	7.9929	7.9931
Baboon	7.7419	7.4631	7.7495	7.9993	7.9993	7.9994
Tumor	7.3179	7.1238	7.0153	7.9996	7.9996	7.9996

Table 6. Comparative analysis of information entropy of Baboon image.

Algorithm	Information entropy			
	R	G	B	Average
Proposed	7.9993	7.9993	7.9994	7.9993
Ref. [49]	7.9993	7.9993	7.9993	7.9993
Ref. [50]	7.9993	7.9992	7.9993	7.9993
Ref. [51]	7.9992	7.9993	7.9991	7.9992
Ref. [52]	7.9993	7.9993	7.9992	7.9993
Ref. [53]	7.9994	7.9991	7.9992	7.9992

where $P(s_i)$ denotes the probability of occurrence of pixel value s_i. Table 5 shows that the information entropy of the ciphertext produced by the proposed encryption algorithm is close to 8. In addition, Table 6 presents a comparison with other algorithms. The results indicate that the proposed algorithm achieves the highest average information entropy for the ciphertext image Baboon. Therefore, the proposed method demonstrates strong cryptographic strength and can effectively resist entropy-based attacks.

4.7. Occlusion attack analysis

Ciphertext images may be subjected to various attacks during transmission, which can lead to partial data loss. Cropping attacks and noise attacks are commonly used to evaluate the robustness of image encryption algorithms. In a cropping attack, a portion of the ciphertext image is removed to simulate data loss during transmission. In a noise attack, random noise is added to the ciphertext image to test the robustness of the encryption scheme.

Figures 8(a)–8(d) illustrate cropping attacks with rates of 1/16 and 1/4, as well as salt-and-pepper noise with densities of 0.01 and 0.1 applied to the ciphertext image of Baboon. Figures 8(e)–8(h) show the corresponding decrypted results. As shown in Fig. 8, although some blurring occurs, the decrypted images remain recognizable. This indicates that the proposed algorithm can effectively mitigate the effects of data loss to a certain extent. Therefore, the proposed scheme demonstrates strong robustness against occlusion attacks.

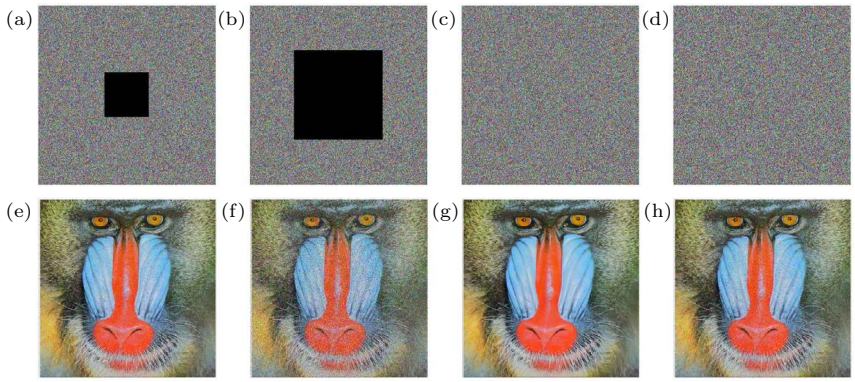


Fig. 8. Occlusion and noise attack analysis. (a) Cropping rate of 1/16. (b) Cropping rate of 1/4. (c) Noise density of 0.01. (d) Noise density of 0.1. (e)–(h) Corresponding decrypted images.

4.8. Sensitivity analysis

Differential attack analysis is a method used to evaluate image encryption systems by introducing slight modifications to one or more pixel values in the plaintext image and observing the resulting changes in the ciphertext image, with the aim of inferring the encryption key or compromising the security of the encryption system. Therefore, an effective image encryption scheme should ensure that even a minor change in the plaintext image results in a completely different ciphertext image. To evaluate the resistance of encryption algorithms against differential attacks, two metrics, namely the unified average changing intensity (UACI) and the number of pixels change rate (NPCR), are commonly used. The formulas for UACI and NPCR are given as follows:

$$\begin{cases} \mathrm{NPCR} = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N} D(i,j)}{M \times N} \times 100\%, \\ \mathrm{UACI} = \dfrac{\sum_{i=1}^{M}\sum_{j=1}^{N} |P_1(i,j) - P_2(i,j)|}{255 \times M \times N} \times 100\%, \end{cases} \tag{5}$$

where

$$D(i,j) = \begin{cases} 0, & P_1(i,j) = P_2(i,j), \\ 1, & P_1(i,j) \neq P_2(i,j). \end{cases}$$

Here, P_1 denotes the encrypted image corresponding to the original plaintext image, and P_2 denotes the encrypted image obtained after randomly modifying the value of a single pixel in the plaintext image. The theoretical values of UACI and NPCR are $\mathrm{NPCR_T} = 99.6094\%$ and $\mathrm{UACI_T} = 33.4635\%$, respectively.

Table 7 lists the NPCR and UACI values for the Eye disease, Pepper, Baboon, and Kumar tumor images across the three color channels. The experimental results show that these values are close to the theoretical expectations, indicating that even a single-pixel change in the plaintext image leads to significant differences in the ciphertext image. To further evaluate the sensitivity of the proposed encryption scheme to plaintext variations, 50 experiments are conducted by randomly selecting a pixel in the plaintext image and modifying its value each time. The resulting NPCR and UACI values are shown in Fig. 9. The results indicate that both NPCR and UACI closely match their theoretical values, demonstrating that the proposed scheme has strong sensitivity to plaintext changes and is effective against differential attacks. Table 8 compares the NPCR and UACI values of different encryption schemes. The results show that the proposed method is closer to the theoretical values, further demonstrating its superiority in resisting differential attacks. Therefore, the proposed encryption scheme exhibits strong robustness and high security against such attacks.

Table 7. The results of NPCR and UACI.

Algorithm	NPCR (%)				UACI (%)			
	R	G	B	Average	R	G	B	Average
Eye disease	99.6231	99.5930	99.6056	99.6072	33.5236	33.4094	33.4098	33.4476
Pepper	99.5960	99.6361	99.6014	99.6112	33.3564	33.5295	33.5152	33.4670
Baboon	99.6132	99.6140	99.6098	99.6132	33.5256	33.4092	33.4678	33.4675
Kumar tumor	99.6062	99.6001	99.6077	99.6047	33.3920	33.4148	33.4340	33.4136

Table 8. Comparative analysis for NPCR and UACI of Baboon image.

Algorithm	NPCR (%)				UACI (%)			
	R	G	B	Average	R	G	B	Average
Proposed	99.6132	99.6140	99.6098	99.6132	33.5256	33.4092	33.4678	33.4675
Ref. [49]	99.6143	99.6246	99.6250	99.6213	33.4566	33.4591	33.4248	33.4468
Ref. [50]	99.6088	99.6115	99.6102	99.6102	33.4705	33.4690	33.4609	33.4668
Ref. [51]	99.6299	99.6037	99.6090	99.6142	33.4236	33.4326	33.4886	33.4483
Ref. [52]	–	–	–	99.6095	–	–	–	33.4564
Ref. [53]	99.6082	99.6078	99.6094	99.6085	33.0672	33.5796	33.6274	33.4247

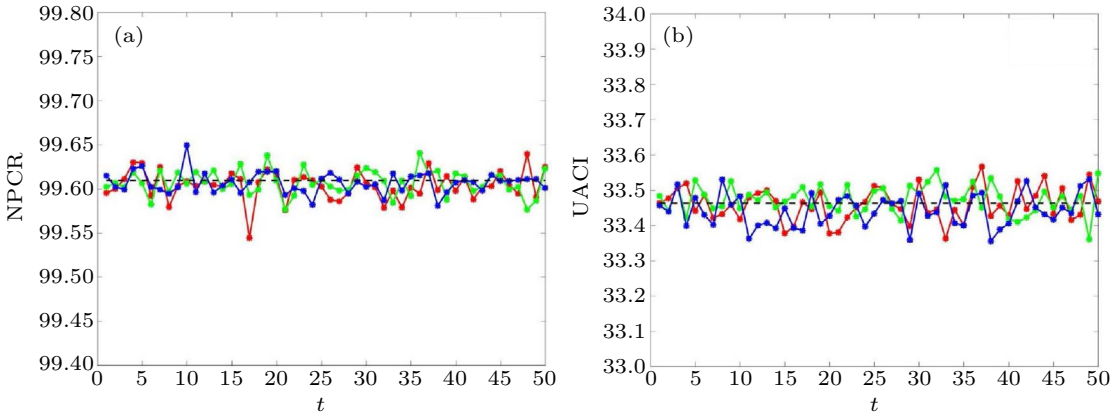


Fig. 9. Randomness test results. (a) NPCR of the Baboon image. (b) UACI of the Baboon image.

4.9. PSNR and MSSIM analysis

In evaluating image encryption performance, the peak signal-to-noise ratio (PSNR) and the mean structural similarity index measure (MSSIM) are widely used as important metrics for assessing the quality and structural similarity between encrypted and original images. PSNR measures the pixel-level differences between the original and encrypted images, while MSSIM evaluates similarity in terms of luminance, contrast, and structural information. A joint evaluation of these two metrics enables both objective and perceptual assessment of the encryption results.

PSNR is commonly used to quantify the degradation in image quality introduced during the encryption process. A higher PSNR indicates greater similarity between the encrypted and original images, implying lower distortion and better visual quality. In contrast, MSSIM focuses on structural similarity rather than pixel-wise differences. It better reflects human visual perception by jointly considering luminance, contrast, and structural similarity. The MSSIM value ranges from 0 to 1, where values closer to 1 indicate higher structural similarity. The formulas for PSNR and MSSIM are given as follows:

$$\begin{cases} \mathrm{PSNR} = 10 \cdot \log_{10} \left\{ \dfrac{\mathrm{MAX}_I^2}{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2} \right\}, \\ \mathrm{MSSIM}(x,y) = \dfrac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \end{cases} \tag{6}$$

where MAX_I denotes the maximum possible pixel value of the image, and m and n denote the height and width of the image, respectively. $I(i,j)$ and $K(i,j)$ represent the pixel values at position (i,j) in the original and encrypted images, respectively. The term $\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$ corresponds to the mean squared error (MSE) between the two images. In the MSSIM formula, μ_x and μ_y denote the mean luminance values of the two images, σ_x^2 and σ_y^2 represent their variances, σ_{xy} denotes their covariance, and c_1 and c_2 are small constants used to ensure numerical stability. PSNR is measured in decibels (dB); generally, values above 30 dB indicate acceptable visual quality. The MSSIM value ranges from 0 to 1, and values closer to 1 indicate better structural similarity.

Figure 10 presents the carrier images Plane, Flowers, Sailboat, and Cablecar and their corresponding stego-images.

Notably, in all cases, the embedded content corresponds to the cipher-image obtained from encrypting the Baboon image. As shown in Table 9, the encrypted images generated by the proposed method achieve high PSNR and MSSIM values, indicating that the embedding process introduces only slight visual distortion while preserving structural information effectively. These results demonstrate that the proposed method maintains both high visual quality and structural consistency, highlighting its effectiveness in practical image encryption and steganography applications.

Table 9. PSNR and MSSIM results for different cover images.

Cover-image	Stego-image	PSNR (dB)	MSSIM
Plane	Fig. 10(e)	44.16	0.9987
Flowers	Fig. 10(f)	44.14	0.9994
Sailboat	Fig. 10(g)	44.15	0.9994
Cablecar	Fig. 10(h)	44.15	0.9992



Fig. 10. Image steganography test results. (a)–(d) Carrier images and (e)–(h) stego-images.

4.10. Time complexity analysis

This section analyzes the time complexity of the proposed algorithm by examining each stage of the encryption process. The analysis considers a medical image of size $M \times N$. First, in the key generation stage, the SHA-256 hash algorithm processes the entire input image of size $M \times N$, resulting in a time complexity of $O(MN)$.

During the pixel scrambling stage, each pixel is accessed once, and its position and channel information are permuted based on chaotic sequences and random indices. Since all operations on each pixel are performed in constant time, the scrambling stage also has a time complexity of $O(MN)$.

Subsequently, in the cellular automaton evolution stage, the state matrix of size $M \times N$ is traversed cell by cell, and each cell is updated using a fixed-size neighborhood with constant-time operations, resulting in a time complexity of $O(MN)$.

In the diffusion stage, DNA encoding, decoding, and logical operations are applied to fixed-length data for each pixel; therefore, the diffusion stage also requires $O(MN)$ time. Finally, in the LSB embedding stage, pixel-wise binary conversion and LSB substitution are performed on both the carrier image and the embedded image. When the embedded image size is $2M \times 2N$, the complexity remains $O(MN)$ in asymptotic terms.

Therefore, the overall time complexity of the proposed encryption algorithm is $O(MN)$, indicating that the algorithm scales linearly with the image size.

5. Conclusion

This study presents a comprehensive and visually meaningful color medical image encryption scheme by integrating three core components: chaotic sequence-driven multi-channel pixel permutation, cellular automaton-based dynamic DNA coding with double-layer diffusion, and chaotic sequence-driven multi-channel LSB embedding, all enhanced by memristive Hopfield neural networks. Extensive experimental results demonstrate that the proposed scheme achieves a sufficiently large key space, high key sensitivity, and strong security performance, as evidenced by low inter-pixel correlation, information entropy close to the theoretical optimum, and NPCR and UACI values near their ideal thresholds. Furthermore, the scheme exhibits strong robustness against noise and data loss attacks, thereby effectively ensuring the confidentiality and integrity of medical image data.

Future work can focus on two directions: first, improving computational efficiency to meet real-time transmission requirements in telemedicine applications; second, exploring the integration of the proposed framework with lightweight cryptographic protocols to enhance its applicability in resource-constrained medical devices.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62202198), the Hunan Natural Science Foundation of China (Grant Nos. 2024JJ7372

and 2022JJ40514), and the Scientific Research Project of the Hunan Provincial Department of Education (Grant No. 24A0550).

References

[1] Banning S, Höglinger M, Meyer D and Reich O 2023 *Telemed. e-Health* **29** 510

[2] Barney A, Buckelew S, Mesheriakova V and Raymond-Flesch M 2020 *J. Adolesc. Health* **67** 164

[3] Wang L, Xu X, Gao S and Mou J 2025 *Chin. Phys. B* accepted

[4] Esteva A, Chou K, Yeung S, *et al.* 2021 *npj Digit. Med.* **4** 5

[5] Kengnou Telem A N, Fotsin H B and Kengne J 2021 *Multimed. Tools Appl.* **80** 19011

[6] Gupta I, Selwal A and Sharma A K 2025 *Comput. Sci. Rev.* **58** 100783

[7] del Moral J O, de Martí i Olius A, Vidal G, Crespo P M and Martinez J E 2024 *IEEE Internet Things J.* **11** 30217

[8] Vasani V, Prateek K, Amin R, Maity S and Dwivedi A D 2024 *J. Ind. Inf. Integr.* **39** 100594

[9] Halunen K and Latvala O M 2021 *Comput. Sci. Rev.* **39** 100340

[10] Zhang Z P, Sun K H, Zhu W T and Wang H H 2025 *Phys. Scr.* **100** 015284

[11] Meng F Q and Wu G 2024 *Expert Syst. Appl.* **254** 124413

[12] Wang Y M, Huang S F, Chen H, Yang J and Cai S T 2024 *Chin. Phys. B* **33** 010502

[13] Hagras T, Salama D and Youness H 2022 *Alex. Eng. J.* **61** 11651

[14] Huo X and Wang X 2023 *Results Eng.* **20** 101589

[15] Ullah S, Liu X, Waheed A and Zhang S 2025 *Comput. Stand. Interfaces* **93** 103980

[16] Yao W, Liu J, Sun Y, *et al.* 2024 *Nonlinear Dyn.* **112** 693

[17] Li X, He Q, Yu T, Cai Z and Xu G 2024 *Chin. Phys. B* **33** 030505

[18] Bencherqui A, Tahiri M A, Karmouni H, *et al.* 2024 *Eng. Sci. Technol. Int. J.* **50** 101612

[19] Wang X and Guan N 2020 *Opt. Laser Technol.* **132** 106501

[20] Liu Y, Sham C W, Zhang F, Zhao H and Ma L 2025 *Opt. Lasers Eng.* **194** 109121

[21] Wang X Y, Wang X L, Teng L, Jiang D H and Xian Y J 2023 *Chin. Phys. B* **32** 020503

[22] Zhang Z, Mou J, Banerjee S and Cao Y 2025 *Chin. Phys. B* **33** 020503

[23] Liu J, Sun K H and Wang H H 2024 *Chaos, Solitons Fractals* **189** 115684

[24] Matthews R 1989 *Cryptologia* **13** 29

[25] Fridrich J 1997 *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, USA, Art. No. 638097

[26] Han M, Zhang R, Qiu T, Xu M and Ren W 2019 *IEEE Trans. Syst., Man, Cybern.: Syst.* **49** 2144

[27] Liu P, Teng L, Liu H, Iu H H C, Wang M and Yan X 2025 *IEEE Internet Things J.* **12** 20087

[28] Ding D, Xie D, Zhang H, Yang Z and Liu C 2025 *IEEE Trans. Circuits Syst. Video Technol.* **35** 6522

[29] Zhou S, Yin Y, Erkan U, Toktas A and Zhang Y 2025 *Chaos Solitons Fractals* **193** 116088

[30] Wang H, Zhu H, Li B, Sun K and Liu W 2024 *Eur. Phys. J. Plus* **139** 826

[31] Ma J 2024 *Appl. Math. Comput.* **463** 128379

[32] Zhou S, Liu H, Iu H H C and Toktas A 2025 *IEEE Internet Things J.* **12** 9082

[33] Liu W, Wang H, Chen Y and Sun K 2024 *Eur. Phys. J. Plus* **139** 473

[34] Yao W, Wang Y, Xiong L, Xiao J, Zhang S and Sun Y 2025 *Nonlinear Dyn* **113** 35237

[35] Zhao S, Zhao L, Wen S and Cheng L 2025 *Artif. Intell. Sci. Eng.* **1** 64

[36] Zhang J, Zhang D, Li D, Wang J and Wang J 2025 *Pattern Recognit.* **167** 111723

[37] Yao W, Gao K, Wang Y, *et al.* 2025 *Neural Netw.* **188** 107502

[38] Ma J 2025 *Nonlinear Dyn.* **113** 25365

[39] He S, Liu J, Wang H and Sun K 2023 *Neurocomputing* **523** 1

[40] Yu F, Zhang S, Su D, Wu Y, Musoya-Yafracia G and Yin H 2025 *Fractal Fract.* **9** 115

[41] Ding D, Xiao H, Yang Z, *et al.* 2022 *Nonlinear Dyn.* **108** 4433

[42] Zhao C, Zhai Z and Zeng B 2025 *Comput. Stand. Interfaces* **93** 103974

[43] Wang X Y, Zhang J J, Zhang F C and Cao G H 2019 *Chin. Phys. B* **28** 040504

[44] Demirkol A S, Ascoli A, Sahin M E, Tetzlaff R, Karakaya B and Ulutas H 2024 *Chaos Solitons Fractals* **183** 114923

[45] Hopfield J J 1984 *Proc. Natl. Acad. Sci. USA* **81** 3088

[46] Deng L, Lyu J, Huang H, Deng Y, Yuan J and Tang X 2020 *Sci. Data* **7** 23

[47] Kumar N, Verma R, Sharma S, Bhargava S, Vahadane A and Sethi A 2017 *IEEE Trans. Med. Imaging* **36** 1550

[48] Ding D, Zhu H, Zhang H, Yang Z and Xie D 2024 *Chaos Solitons Fractals* **185** 115168

[49] Wang M, Teng L, Zhou W, Yan X, Xia Z and Zhou S 2025 *Expert Syst. Appl.* **261** 125328

[50] Toktas F, Erkan U and Yetgin Z 2024 *Expert Syst. Appl.* **249** 123583

[51] Xin J, Hu H and Zheng J 2023 *Nonlinear Dyn.* **111** 7859

[52] Gao S, Zhang Z, Mou J, *et al.* 2025 *IEEE Internet Things J.* **12** 18115

[53] Cheng T, Zhao R S, Wang S, Wang K H and Ma H Y 2025 *Chin. Phys. B* **34** 010305