

Resonant tunneling diode cellular neural network with memristor coupling and its application in police forensic digital image protection

Fei Yu(余飞), Dan Su(苏丹), Shaoqi He(何邵祁), Yiya Wu(吴亦雅), Shankou Zhang(张善扣), and Huige Yin(尹挥戈)

Citation: Chin. Phys. B, 2025, 34 (**5**): 050502. DOI: 10.1088/1674-1056/adb8bb
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

Study and circuit design of stochastic resonance system based on memristor chaos induction

Qi Liang(梁琦), Wen-Xin Yu(于文新), and Qiu-Mei Xiao(肖求美)
Chin. Phys. B, 2025, 34 (**4**): 040502. DOI: 10.1088/1674-1056/adb268

Dynamics and synchronization of neural models with memristive membranes under energy coupling

Jingyue Wan(万婧玥), Fuqiang Wu(吴富强), Jun Ma(马军), and Wenshuai Wang(汪文帅)
Chin. Phys. B, 2024, 33 (**5**): 050504. DOI: 10.1088/1674-1056/ad1dcc

Fractional-order heterogeneous memristive Rulkov neuronal network and its medical image watermarking application

Dawei Ding(丁大为), Yan Niu(牛炎), Hongwei Zhang(张红伟), Zongli Yang(杨宗立), Jin Wang(王金), Wei Wang(王威), and Mouyuan Wang(王谋媛)
Chin. Phys. B, 2024, 33 (**5**): 050503. DOI: 10.1088/1674-1056/ad322b

Dynamics analysis and cryptographic implementation of a fractional-order memristive cellular neural network model

Xinwei Zhou(周新卫), Donghua Jiang(蒋东华), Jean De Dieu Nkapkop, Musheer Ahmad, Jules Tagne Fossi, Nestor Tsafack, and Jianhua Wu(吴建华)
Chin. Phys. B, 2024, 33 (**4**): 040506. DOI: 10.1088/1674-1056/ad03dd

Dynamical analysis, geometric control and digital hardware implementation of a complex-valued laser system with a locally active memristor

Yi-Qun Li(李逸群), Jian Liu(刘坚), Chun-Biao Li(李春彪), Zhi-Feng Hao(郝志峰), and Xiao-Tong Zhang(张晓彤)
Chin. Phys. B, 2023, 32 (**8**): 080503. DOI: 10.1088/1674-1056/acd68b

---

Resonant tunneling diode cellular neural network with memristor coupling and its application in police forensic digital image protection

Fei Yu(余飞)[†], Dan Su(苏丹), Shaoqi He(何邵祁), Yiya Wu(吴亦雅), Shankou Zhang(张善扣), and Huige Yin(尹挥戈)

School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, *China*

(Received 16 December 2024; revised manuscript received 6 February 2025; accepted manuscript online 21 February 2025)

Due to their biological interpretability, memristors are widely used to simulate synapses between artificial neural networks. As a type of neural network whose dynamic behavior can be explained, the coupling of resonant tunneling diode-based cellular neural networks (RTD-CNNs) with memristors has rarely been reported in the literature. Therefore, this paper designs a coupled RTD-CNN model with memristors (RTD-MCNN), investigating and analyzing the dynamic behavior of the RTD-MCNN. Based on this model, a simple encryption scheme for the protection of digital images in police forensic applications is proposed. The results show that the RTD-MCNN can have two positive Lyapunov exponents, and its output is influenced by the initial values, exhibiting multistability. Furthermore, a set of amplitudes in its output sequence is affected by the internal parameters of the memristor, leading to nonlinear variations. Undoubtedly, the rich dynamic behaviors described above make the RTD-MCNN highly suitable for the design of chaos-based encryption schemes in the field of privacy protection. Encryption tests and security analyses validate the effectiveness of this scheme.

Keywords: memristor, hyperchaos, resonant tunneling diode-based cellular neural network (RTD-CNN), dynamic analysis, image encryption

PACS: 05.45.–a, 05.45.Gg, 05.45.Jn, 05.45.Vx

DOI: 10.1088/1674-1056/adb8bb **CSTR:** 32038.14.CPB.adb8bb

1. Introduction

Simulating the human brain has long been a goal in the field of computer biomimicry. The brain contains a vast number of interconnected neurons that interact to produce a rich array of dynamical behaviors, enabling humans to make efficient and accurate judgments in daily life.[1–5] Since the introduction of the Hodgkin–Huxley neuron model in 1952,[6] numerous new neuron models and neural network models have emerged. Neuron models, such as the Hindmarsh–Rose model and the Morris–Lecar model, primarily explain the phenomenon of bursting firing in neurons.[7,8] Neural network models including the cellular neural network (CNN) and Hopfield neural network (HNN), simulate various working and learning patterns of the human brain and other biological systems to achieve intelligence, exhibiting complex dynamical behaviors such as chaos and period.[9,10] Therefore, they have broad application prospects in fields such as sentiment analysis, signal processing, pattern recognition and proportional–integral–derivative controllers.[5,11–17] Itoh *et al.* coupled memristors with CNNs, demonstrating the potential application value of memristive CNNs (MCNNs) in image processing.[18] Furthermore, the memory characteristics of memristors can serve as an adaptive function in feedback circuits.[19] Additionally, systems coupled with memristors can exhibit more diverse and complex dynamic behaviors. Notably, Itoh *et al.* also proposed a resonant tunneling diode-based cellular neural network (RTD-CNN) model and validated the associative memory characteristics of this model in their research.[20] However, the current dynamic description of RTD-CNN is still incomplete and many potential dynamical behaviors await further exploration.

A memristor is a nonlinear electronic component proposed by Chua in 1971.[21] It establishes the relationship between magnetic flux and electric charge, distinguishing it as the fourth fundamental electronic component, alongside resistors, inductors and capacitors. Due to its operating principle, it is widely used to simulate synaptic weights in neural network models and in external electromagnetic radiation applications.[22–29] The nonlinearity, non-volatility and local active characteristics of memristors lead to rich dynamic behaviors in neural network models that couple with them.[30–34] In recent years, the coupling of memristors with neural networks has continued. In the simplest HNN model with only two neurons, Lin *et al.* replaced the connection weights between different neurons and the self-synaptic weights with memristors, adding a memristor to simulate external electromagnetic radiation on one of the neurons and discovering the dynamic phenomenon of multi-butterfly attractors.[35] Wang *et al.* used discrete memristors to couple CNN models with heterogeneous activation functions, revealing dynamic behaviors such as hyperchaos and transient chaos within Ref. [36]. In addition, Qi *et al.* made a preliminary study of the dynamic behavior of MCNNs.[37] The research by Wu *et al.* and the paper by Chen *et al.* explored the synchronization behaviors of

[†]Corresponding author. E-mail: yufeiyfyf@csust.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

MCNNs.[38,39]

Memristor neural networks have significant application value in the fields of information transmission and data encryption, attracting widespread attention from researchers.[40–48] An ideal random sequence for data security should possess high randomness and ergodicity, and chaotic systems, due to their sensitivity to initial conditions and unpredictability, can generate sequences that meet this requirement well.[49–51] In recent years, researchers have designed various encryption schemes based on chaotic systems for different application scenarios, such as medical privacy protection[52] and industrial Internet of things (IoT).[53,54] From the perspective of engineering applications, encryption algorithms should strive to maintain simplicity, a fast encryption speed and high reliability. Compared with the widely used memristor Hopfield neural network (MHNN), MCNN does not require the use of hyperbolic tangent functions, making it easier to implement and providing certain advantages in chaotic sequence generation speed. Furthermore, studies have shown that can easily enter a hyperchaotic state, as validated in the research by Chen[55] and Huang.[56] However, these studies present certain areas for improvement: on the one hand, there is an opportunity for further exploration into how system parameters influence chaotic states, which could enhance system tunability; on the other hand, the research does not fully address the examination of the system's multistability characteristics. In fact, hyperchaotic systems are generally considered to have greater complexity than ordinary chaotic systems,[57,58] thus demonstrating superior performance in encryption applications.[59–62] Inspired by the aforementioned studies, this paper proposes a hyperchaotic resonant tunneling diode-based cellular neural network with memristor (RTD-MCNN) system with multistable characteristics. Compared with existing research, this system exhibits more complex dynamical behavior influenced by the internal parameters of the memristor. Additionally, the encryption scheme designed in this paper extends the key length by 1.2 times, achieving better encryption effects and further enhancing the system's potential applications in data security.

The main contributions of this paper are divided into three parts as follows: (1) We designed a memristor model incorporating an absolute value function and coupled it to a CNN by replacing the connection weights between cells, resulting in the RTD-MCNN model that exhibits rich dynamic behaviors. (2) Through numerical calculations, we revealed the hyperchaotic and multistable dynamic behaviors of the RTD-MCNN model. The internal parameters of the memristor control whether the attractors produced by the RTD-MCNN are periodic, chaotic or hyperchaotic, and a set of amplitudes of the attractors shows a smooth nonlinear correlation with these parameters. (3) For the scenario of police image protection, we developed an encryption scheme based on a RTD-MCNN, with numerical validation demonstrating its satisfactory reliability.

The remainder of this paper is structured as follows. Section 2 introduces the construction of the RTD-MCNN, including the design of the memristor model. Section 3 reveals the dynamic behaviors such as hyperchaos and multistability of the model. Section 4 presents a simple encryption scheme, including necessary randomness verification methods and an evaluation of the encryption performance. Section 5 concludes the paper.

2. RTD-MCNN model design

2.1. Memristor design

According to Ref. [63], a memristor establishes a relationship between electric charge and magnetic flux. This paper presents a novel memristor model that can be expressed by the following equation:

$$\begin{cases} i = W(\phi)v_{\mathrm{m}} = (1-0.25|\phi|^3)v_{\mathrm{m}}, \\ \dfrac{\mathrm{d}\phi}{\mathrm{d}t} = 2.5v_{\mathrm{m}} - k\phi, \end{cases} \tag{1}$$

where v_{m} and i represent the input voltage and output current of the memristor, respectively, ϕ denotes the internal state variable of the memristor and k is an arbitrary real number. $W(\phi)$ represents the memristor's memductance. When an external periodic bipolar voltage stimulus is applied, the i–v_{m} characteristic of the memristor exhibits an '8'-shaped curve, which corresponds to the memristor's pinched hysteresis loop (PHL). In this paper, the voltage signal $v_{\mathrm{m}} = A\sin(2\pi ft)$ is used as the input voltage for the memristor. When $k = 0.5$, the differential Eq. (1) is solved using the ode45 function in MATLAB 2021b to simulate the PHLs corresponding to different voltage amplitudes A and voltage signal frequencies f, as shown in Fig. 1.

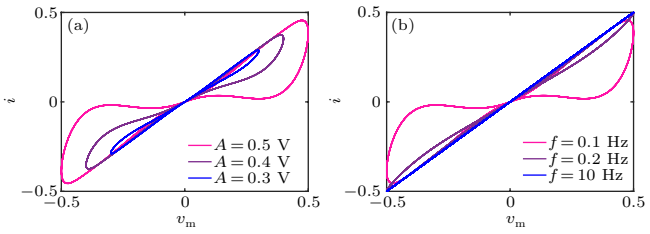


Fig. 1. The PHLs of the memristor model 1 under different parameters: (a) PHLs with different voltage amplitudes for $f = 0.1$ Hz; (b) PHLs at different voltage frequencies for $A = 0.5$ V.

Figure 1(a) shows that with a fixed input voltage frequency, the area of the side lobes enclosed by the memristor's hysteresis loop increases as the amplitude of the input voltage rises. In contrast, figure 1(b) indicates that with a fixed input voltage amplitude, the side lobe area decreases as the input

voltage frequency increases, and when the frequency is sufficiently high the hysteresis loop approaches a nearly single-valued function. Additionally, in both cases, the hysteresis loops are compressed at the origin. From the above conclusions, it can be inferred that the model proposed in this paper satisfies the definition of a memristor.

2.2. RTD-CNN based on a memristor

CNNs are models that can be explained through dynamic behavior and were first proposed by Chua in 1988.[9] They are a type of neural network model structured as a two-dimensional network, consisting of $M \times N$ neuron cells. Adjacent cells are directly connected, while non-adjacent cells are indirectly connected through propagation effects. Figure 2 illustrates the topology of a 4×4 CNN.

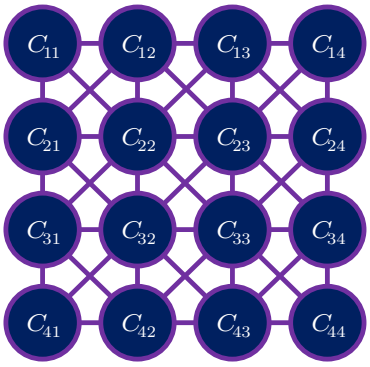


Fig. 2. The topology of a 4×4 CNN.

The state equaiton for each neuron cell C_{ij} where $i \in M$ and $j \in N$ can be represented by the following first-order differential equation:

$$C\frac{\mathrm{d}x_{ij}(t)}{\mathrm{d}t} = -\frac{x_{ij}(t)}{R_x} + \sum_{C_{kl}\in N_r(i,j)} A(i,j:k,l)y_{kl}(t) + \sum_{C_{kl}\in N_r(i,j)} B(i,j:k,l)u_{kl}(t) + I, \tag{2}$$

where x_{ij} represents the current state of the neuron cell, u_{kl} denotes the output of neighboring cells, A_{ij} is the feedback module and B_{ij} is the control module. A_{ij} and B_{ij} represent the connection weights between the outputs and inputs of neighboring cells and the neuron cell, respectively. C is the linear capacitance, I is the input bias current and R is the output impedance. The nonlinear activation function $y_{kl}(t)$ can be expressed by the following equation:

$$y_{kl} = f(x_{kl}(t)) = 0.5(|x_{kl}(t)+1| - |x_{kl}(t)-1|). \tag{3}$$

The characteristics of local interconnections and the nonlinear activation functions give CNNs a rich variety of dynamic behaviors. In Ref. [55], Chen *et al.* proposed a based on traditional CNNs. The expression was presented as

$$\begin{cases} \dot{x}_1 = -(x_3 + x_4), \\ \dot{x}_2 = 2x_2 + x_3, \\ \dot{x}_3 = 14(x_1 - x_2), \\ \dot{x}_4 = 100(x_1 - g(x_4) + w), \end{cases} \tag{4}$$

where w is a constant. The function $g(x_4)$ is defined as $g(x_4) = \alpha x_4 - (|x_4 - 0.4| - |x_4 - 0.8| - |x_4 + 0.4| + |x_4 + 0.8|)$.

Unfortunately, the dynamics observed in Ref. [55] were somewhat limited, primarily concentrating on the effects of parameter variations within the system. As a special electronic component, memristors can well simulate the connection synapse between neuronal cells.[64,65] Inspired by these, we also incorporate a memristor into the connection circuits between neurons so that the memristor's memductance becomes the weight coefficient of the synapse,[66] and a five-dimensional RTD-MCNN is obtained as

$$\begin{cases} \dot{x_1} = -(x_3 + x_4), \\ \dot{x_2} = 2W(\phi)x_2 + x_3, \\ \dot{x_3} = 14(x_1 - x_2), \\ \dot{x_4} = 100(x_1 - g(x_4)), \\ \dot{\phi} = 2.5x_2 - k\phi, \end{cases} \tag{5}$$

where $W(\phi) = 1 - 0.25|\phi|^3$ represents the memductance of the memristor, $|\phi|$ is the state variable, x_i for $i \in \{1,2,3,4\}$ denotes the state variables of the four neuron cells, k is a constant and $\alpha = 1$.

2.3. Equilibrium point characteristic

To gain a deeper understanding of the dynamic behavior of the RTD-MCNN, this section will explore the distribution and stability of the equilibrium points of Eq. (5) through both theoretical and numerical analysis. By setting the left side of Eq. (5) to zero, we can solve for the equilibrium points of the system, which can be expressed as follows:

$$\begin{cases} 0 = -(x_3 + x_4), \\ 0 = 2W(\phi)x_2 + x_3, \\ 0 = 14(x_1 - x_2), \\ 0 = 100(x_1 - g(x_4)), \\ 0 = 2.5x_2 - k\phi. \end{cases} \tag{6}$$

Furthermore, the Jacobian matrix of Eq. (5) can be obtained as follows:

$$J = \begin{bmatrix} 0 & 0 & -1 & -1 & 0 \\ 0 & 2W(\phi) & 1 & 0 & 2W'(\phi)x_2 \\ 14 & -14 & 0 & 0 & 0 \\ 100 & 0 & 0 & -100g'(x_4) & 0 \\ 0 & 2.5 & 0 & 0 & -k \end{bmatrix}, \tag{7}$$

where $W(\phi) = 1 - 0.25\phi^2|\phi|$ and $W'(\phi) = -\frac{3}{4}\phi^2\mathrm{sign}(\phi)$. The derivative $g'(x_4)$ is given by

$$g'(x_4) = \big(1 - \mathrm{sign}(x_4 - 0.4) + \mathrm{sign}(x_4 + 0.4) + \mathrm{sign}(x_4 - 0.8) - \mathrm{sign}(x_4 + 0.8)\big).$$

From Eq. (6), the relationships between the variables can be easily derived as $x_1 = x_2 = (2k/5)\phi$, $x_3 = -x_4$. Thus, equation (8) can be listed as follows:

$$\begin{cases} x_3 = -2W(\phi)x_2, \\ x_1 = g(x_4), \end{cases} \Rightarrow \begin{cases} H_1 : 5x_4 - 4k\phi W(\phi) = 0, \\ H_2 : 2k\phi - 5g(x_4) = 0. \end{cases} \tag{8}$$

Table 1. Equilibrium points, eigenvalues and types of equilibrium points for a RTD-MCNN at $k = 0.5$.

Number	Equilibrium points	Eigenvalues	Type
E_1	$-(0.2520, 0.2520, -0.2520, 0.2520, 1.2599)$	$-98.9912, 0.0164 \pm 5.2816\mathrm{i}$ $-0.2707 \pm 0.8290\mathrm{i}$	USF-1
E_2	$-(0, 0, 0, 0)$	$-98.9912, -0.5000, 0.5456$ $-0.2224 \pm 5.0826\mathrm{i}$	USF-2
E_3	$(0.2520, 0.2520, -0.2520, 0.2520, 1.2599)$	$-98.9912, 0.0164 \pm 5.2816\mathrm{i}$ $-0.2707 \pm 0.8290\mathrm{i}$	USF-1

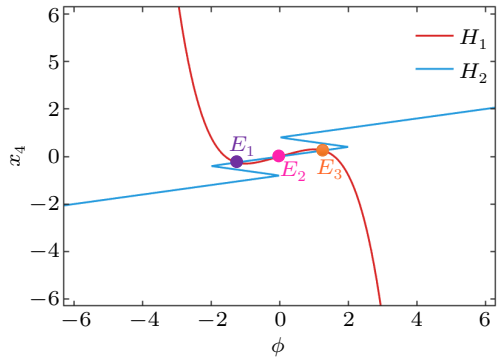


Fig. 3. The intersection graph of curves H_1 and H_2 for $k = 0.5$.

By solving Eq. (8), the equilibrium points of model (5) can be determined. For example, to fix the parameter $k = 0.5$, equation (8) contains only the independent variables ϕ and x_4. The plots of H_1 and H_2 can be obtained, as shown in Fig. 3. Clearly, the equilibrium point is the intersections of the two curves H_1 and H_2 in Eq. (8). Furthermore, the built-in implicit function solver fsolve in MATLAB can be used to obtain numerical solutions. Besides, substituting these solutions into the Jacobian matrix J allows for the calculation of the eigenvalues at the equilibrium points. Table 1 presents the properties of the equilibrium points under different parameter conditions. Here USF-1 and USF-2 represent the unstable saddle focus of index 1 and index 2, respectively.

The equilibrium points of a dynamical system characterize the stability of the system in the vicinity of those points. As shown in Table 1, when appropriate internal parameters are selected, the equilibrium points of the system exhibit instability, a characteristic that satisfies the fundamental conditions for chaos. Specifically, for the equilibrium points E_1 and E_3, the eigenvalues can be taken as $\gamma = -98.9912$ and $\delta \pm jw = 0.0164 \pm 5.2816\mathrm{i}$. For the equilibrium point E_2, the eigenvalues can be taken as $\gamma = -98.9912$ and $\delta \pm jw = -0.2224 \pm 5.0826\mathrm{i}$. These eigenvalues all satisfy $|\gamma \times \delta| < 0$ and $|\delta|/|\gamma| < 1$. Therefore, according to Shilnikov's theorem, it can be concluded that Eq. (5) exhibits chaotic behavior characterized by Smale's horseshoe mapping, indicating that the equilibrium points are unstable.

3. Dynamic behavior analysis

In this section, a dynamic analysis of the proposed RTD-MCNN will be conducted. The tools required for this analysis include bifurcation diagrams, Lyapunov exponents, phase portraits, time series and basins of attraction. It is important to note that the experimental platform used is MATLAB 2021b. The differential equations and Lyapunov exponents are computed using the ode45 function and Wolf's method, respectively. Unless specified otherwise, a time step of 0.01 is used, with a total simulation time of 2000.

3.1. The k-relied dynamical behaviors

To further investigate the impact of the memristor on the RTD-CNN this subsection considers the effect of the internal parameter k of the memristor on the entire neural network system. The initial condition (IC) is set to $[0.1, 0.1, 0.1, 0.1, 0.1]$. Within the range of $k \in [0, 2]$, the corresponding bifurcation diagram and Lyapunov exponents are presented in Figs. 4(a) and 4(b), respectively. The shape and density of the bifurcation diagram reflect the dynamical behavior of the system. The Lyapunov exponent serves as a criterion for determining whether the system is chaotic: a positive exponent indicates chaos, while the presence of two or more positive exponents suggests hyperchaos. Hyperchaotic systems are generally considered to exhibit greater complexity and richer dynamical behaviors than chaotic systems, providing enhanced reliability for digital image encryption outcomes.

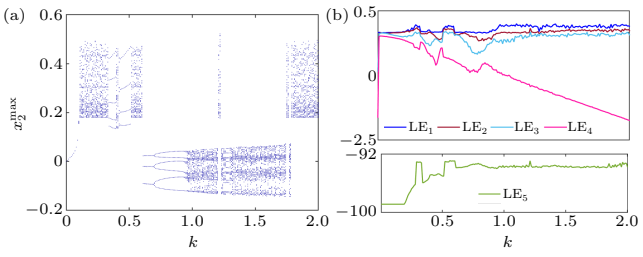


Fig. 4. The bifurcation diagram and Lyapunov exponent spectrum corresponding to the initial condition $[0.1, 0.1, 0.1, 0.1, 0.1]$ and the internal parameter $k \in [0, 2]$: (a) bifurcation diagram; (b) Lyapunov exponent spectrum.

Based on the simulation results, the bifurcation diagram and the Lyapunov exponents spectrum are highly consistent, indicating the accuracy of the simulation outcomes. Further analysis of Figs. 4(a) and 4(b) reveals that within the range of $k \in [0, 0.1]$, the RTD-MCNN is in a periodic state, with the bifurcation diagram represented as a single curve and no Lyapunov exponents greater than zero. At $k = 0.1$, figure 4(a) shows a chaotic bifurcation, with the diagram be-

coming denser, while figure 4(b) reveals a Lyapunov exponent greater than zero. As the value of k increases, two positive exponents emerge at $k = 0.29$, indicating that the system enters a hyperchaotic state. Similarly, when $k \in [0.52, 0.60] \cup [1.8, 2.0]$, the system remains in a hyperchaotic state, characterized by two positive Lyapunov exponents and a bifurcation diagram indicating chaotic bifurcation. In the range of $k \in [1.0, 1.8]$, the system exhibits one positive Lyapunov exponent, indicating that the RTD-MCNN is in a chaotic state. Crisis situations appear in the bifurcation diagram at $k = 0.33$ and $k = 0.60$: referring to the Lyapunov exponent spectrum, these points mark the onset of periodic windows. After $k = 0.6$, the pathway into chaos is observed as a forward doubling bifurcation. Additionally, during the evolution of the parameter k, the bifurcation diagram exhibits multiple jumps, showcasing a rich array of dynamical behavior.

To verify the above analysis, figure 5 presents the phase portraits and time series corresponding to different values of k. The results indicate that when $k = 0.1$ the phase portrait exhibits a "circular" shape, and the amplitude of the time series gradually increases over time, indicating that the RTD-MCNN behaves as an unstable limit cycle. When $k = 0.25$, the phase portrait shows disordered yet bounded characteristics, corresponding to a chaotic attractor, with the time domain waveform displaying an unordered state. For $k = 0.5$, the system corresponds to a stable limit cycle, reflecting a periodic state where the amplitude of the time domain waveform stabilizes over time. Figures 5(d1) and 5(d2) correspond to the phase portrait and time series when $k = 1.0$, displaying chaotic behavior with an unordered time domain waveform; moreover, the phase portrait is asymmetric around the origin compared with Fig. 5(b1). Notably, the attractors described in Figs. 5(b1) and 5(d1) both contain an unstable zero equilibrium point within their basins of attraction. Additionally, apart from the origin, the other two equilibrium points are symmetric with respect to the origin. Subsequent sections will discuss the attractors that exhibit symmetry around the origin, all of which include an unstable equilibrium point. Therefore, the chaotic attractors described in Fig. 5 are self-excited.

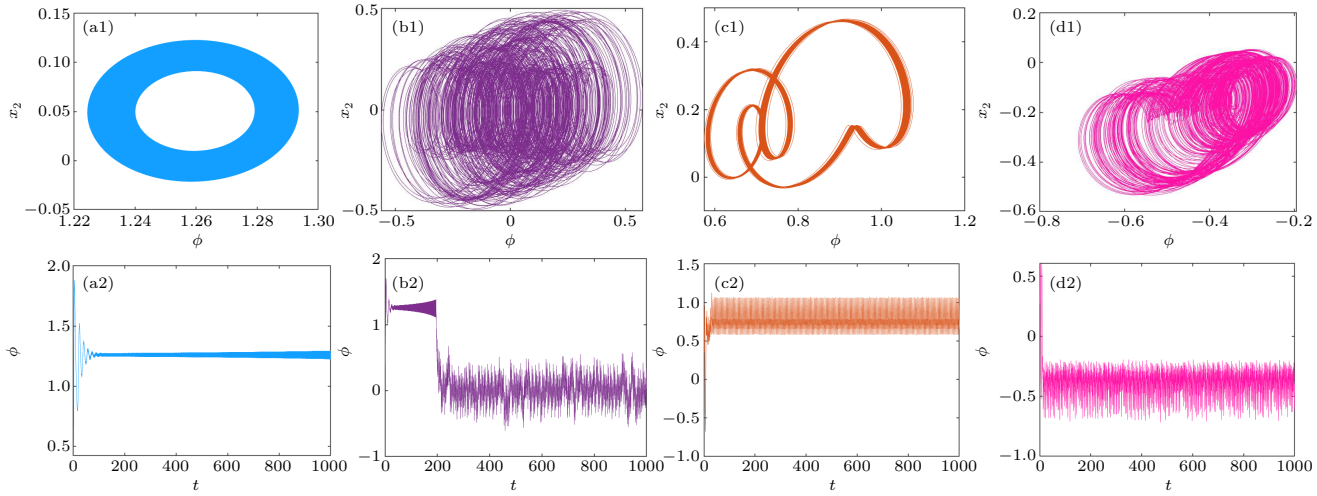


Fig. 5. Phase and time series corresponding to different values of k: (a1) phase with $k = 0.1$; (a2) time series with $k = 0.1$; (b1) phase with $k = 0.25$; (b2) time series with $k = 0.25$; (c1) phase with $k = 0.5$; (c2) time series with $k = 0.5$; (d1) phase with $k = 1.0$; (d2) time series with $k = 1.0$.

3.2. Multistability

In dynamic systems, choosing different initial conditions leads to variations in the phase portraits of the system in terms of space, size and shape; this phenomenon is known as attractor coexistence. In a chaotic image encryption system, the initial values typically serve as keys during the encryption and decryption processes, with different initial values corresponding to completely different chaotic sequences. Furthermore, when a system has three coexisting attractors, it can be referred to as a multistable dynamic system. Therefore, in the field of image encryption, attractor coexistence is a highly beneficial phenomenon, as different initial values correspond to different chaotic attractors, significantly increasing the complexity of the chaotic sequences. As discussed above, as the parameter k evolves, the RTD-MCNN exhibits a rich variety of dynamical behaviors. Notably, the jumps in the bifurcation diagram and the asymmetric distribution in the phase portrait may indicate the dynamic phenomenon of attractor coexistence. Fortunately, this study reveals the multistability of the RTD-MCNN, characterized by the coexistence of three attractors.

When fixing the internal parameter of the memristor at $k = 1.3$ and setting the initial values to $[-0.2, 0.3, 0.2, 0.1, 0.1]$, $[-0.1, 0.1, 0.1, -0.1, -0.1]$ and $[0.2, 0.3, 0.2, 0.1, 0.1]$, the RTD-MCNN generates three distinct attractors. These initial values correspond to attractor 1 (blue), attractor 2 (yellow) and attractor 3 (orange), as shown in Fig. 6(a1). Figure 6(a2) presents the time domain representation of the parameter φ, highlighting the spatial differences between the three attractors. Furthermore, the Lyapunov exponents under different

initial conditions are calculated and presented in Table 2, indicating that attractor 2 is hyperchaotic while attractors 1 and 3 are chaotic. This demonstrates the coexistence of hyperchaotic and chaotic attractors. Similarly, when $k = 0.5$ is set without changing other conditions, the phenomenon of coexistence between hyperchaos and periodic behavior can be observed, with the phase portrait and time domain representations shown in Figs. 6(b1) and 6(b2). Table 2 also provides the corresponding Lyapunov exponents, confirming the coexistence of hyperchaotic and periodic behavior.

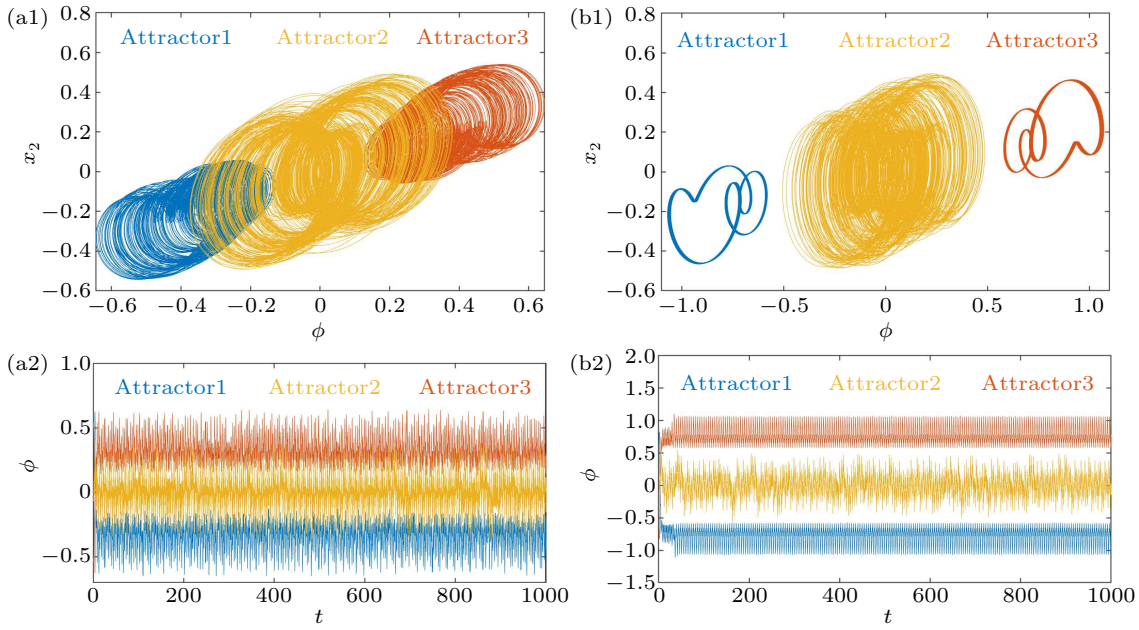


Fig. 6. The phase portraits and time series of the RTD-MCNN under different parameter settings: (a1) phase portrait for $k = 1.3$; (a2) time series for $k = 1.3$; (b1) phase portrait for $k = 0.5$; (b2) time series for $k = 0.5$.

Table 2. The Lyapunov exponents and dynamical behavior under different parameter settings.

Internal parameters k and initial conditions		Lyapunov exponent					Dynamic behavior
Initial conditions	k	LE_1	LE_2	LE_3	LE_4	LE_5	
$[-0.1, 0.1, 0.1, -0.1, -0.1]$	$k = 1.3$	0.1435	0.0787	0.0027	−1.3055	−93.0547	hyperchaos
	$k = 0.5$	0.1393	0.0882	−0.0014	−0.5049	−93.0037	hyperchaos
$[0.2, 0.3, 0.2, 0.1, 0.1]$	$k = 1.3$	0.0924	−0.0020	−0.0441	−1.2769	−93.8922	chaos
	$k = 0.5$	0.0017	−0.1599	−0.2691	−0.2872	−95.2333	non-chaos
$-[0.2, 0.3, 0.2, 0.1, 0.1]$	$k = 1.3$	0.1171	0.0041	−0.0423	−1.2752	−93.0037	chaos
	$k = 0.5$	0.0015	−0.1608	−0.2688	−0.2882	−95.2317	non-chaos

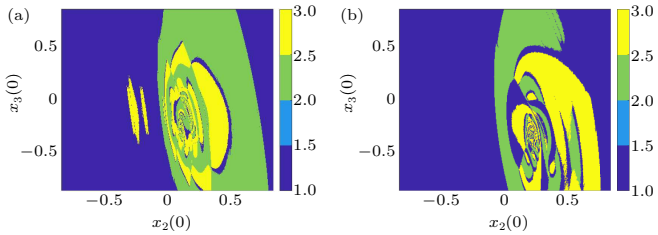


Fig. 7. Basins of attraction corresponding to different ICs and k values: (a) ICs $= [0.1, x_2(0), x_3(0), 0.1, 0.1]$ and $k = 1.3$; (b) ICs $= [0.2, x_2(0), x_3(0), 0.1, 0.1]$ and $k = 0.5$.

In addition, this paper discusses how the basin of attraction reflects the influence of multiple initial value variables on the distribution of attractors. When the initial values are set as $[0.1, x_2(0), x_3(0), 0.1, 0.1]$, with $x_2(0)$ and $x_3(0)$ ranging from $[-0.8, 0.8]$, the basin of attraction is illustrated in Fig. 7(a) when $k = 1.3$. The selected initial values in the green, purple and yellow regions correspond to attractors 1, 2 and 3, respectively, as shown in Fig. 6(a). The results indicate that different colors are widely distributed within the basin of attraction, demonstrating the universality of coexistence phenomena. Similarly, when the initial values are set as $[0.2, x_2(0), x_3(0), 0.1, 0.1]$ and $k = 0.5$, the basin of attraction is presented in Fig. 7(b). Figures 7(a) and 7(b) illustrate that the distribution of coexisting attractors can be controlled by the internal parameters of the memristor.

3.3. Amplitude control

Amplitude control can increase the parameters that control chaos, thereby extending the key length of the random number generator and enhancing the security and reliability of encryption. Some literature achieves the effect of amplitude control by introducing external functions. Interestingly, the RTD-MCNN can achieve partial amplitude control of the sys-

tem simply by adjusting the internal parameters of the memristor. Compared with existing methods of chaotic amplitude control, this approach significantly reduces the complexity of the system without compromising the effectiveness of encryption, facilitating lightweight implementation.

In this experiment, the initial values are fixed at $[-0.1, 0.1, 0.1, -0.1, -0.1]$ and the internal parameters of the memristor are set to $k \in [1, 4]$. Different attractors can be obtained, as shown in Fig. 8(a). As the value of k gradually increases, the attractors exhibit a trend of compression in the direction of ϕ. To provide a more comprehensive and intuitive understanding of the changes in amplitude, figures 8(b) and 8(c) present the maximum amplitudes of the attractors in multiple directions corresponding to different k values. The results indicate that as the value of k increases from 1 to 4, the maximum amplitude in the ϕ direction remains nearly constant, while ϕ shows a nonlinear decreasing trend. This demonstrates that the parameter k indeed has a controlling effect on the amplitude of the attractors.

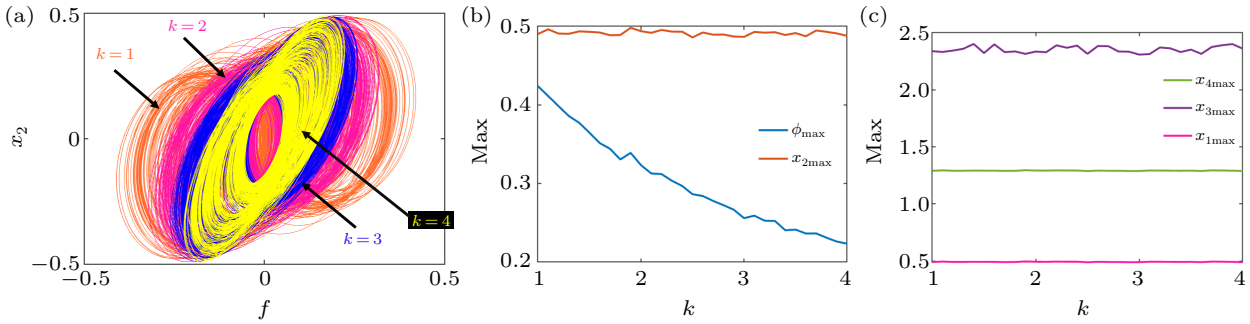


Fig. 8. Amplitude control using parameter k. (a) The phase diagram corresponding to different parameters k. (b) The maximum amplitude varying with the parameter k.

4. Image encryption

In recent years, chaotic sequences have garnered significant attention due to their superior randomness. Utilizing chaotic sequences as random sequences in digital image encryption results in encrypted images that exhibit excellent uniformity in pixel value distribution and statistical characteristics. Furthermore, chaotic digital image encryption benefits from a key space composed of floating-point arrays, rendering brute-force attacks nearly impossible. This technology has demonstrated considerable application value in various fields, including industrial IoT, medical privacy protection and digital image encryption.[67–73] In fact, at critical crime scenes, law enforcement officers are required to conduct evidence collection that may involve sensitive information such as fingerprints, environmental conditions and footprints. These details are often backed up as digital images; if accessed by criminals, they could not only compromise personal privacy but also severely hinder the investigators' ability to solve cases. Given the volume and time sensitivity of physical evidence images, there is a pressing need for a lightweight and reliable encryption scheme to protect these sensitive images. Based on the RTD-MCNN proposed in this paper, a simple and highly reliable encryption scheme is designed for digital image protection in police forensic applications.

4.1. Encryption algorithm

A typical chaotic digital image encryption algorithm comprises two primary modules: permutation and diffusion of the plaintext image. The frequency of permutation and diffusion varies across different encryption algorithms. This study utilizes two distinct sets of chaotic sequences generated by the RTD-MCNN during the permutation and diffusion steps. The encryption process is as follows.

Step 1 Let the original image P consist of $M \times N$ pixels. The initial parameters and values are passed as the initial key to the RTD-MCNN. Using a fourth-order Runge–Kutta algorithm with a step size of 0.01, five chaotic values $(x_1(i), x_2(i), x_3(i), x_4(i), \phi(i))$ are iteratively generated. During this operation, the first 500 unstable jitter points from the system are discarded.

Step 2 The sequences used for permutation and diffusion are denoted as S_1 and S_2, respectively. These sequences are derived from several of the chaotic values $(x_1(i), x_2(i), x_3(i), x_4(i), \phi(i))$ for $i \geq 500$, arranged according to different permutation rules, as described in Eq. (9). Subsequently, the floating-point numbers must be converted into the GF(256) values required for the encryption operations, as outlined below:

$$\begin{cases} S_1(3\times k+1:3\times k+3) &= \{x_1(k), x_2(k), x_3(k)\}, \\ S_2(3\times k+1:3\times k+3) &= \{x_2(k), x_4(k), \phi(k)\}, \end{cases} k \geq 0, \quad (9)$$

$$\begin{cases} S_1 = \mathrm{mod}\,(\mathrm{floor}(S_1 \times 2^{16}), 256), \\ S_2 = \mathrm{mod}\,(\mathrm{floor}(S_2 \times 2^{16}), 256), \end{cases} \quad (10)$$

where floor$(\cdot)$ is the round down function.

Step 3 Using the random sequence S_1, a unique pseudo-random sequence S_3 of length $M \times N$ is generated, and the

original image P undergoes a permutation operation as described in Eq. (11) to produce P_1:

$$P_1(i) = P(\text{index}(S_3(i))). \tag{11}$$

Step 4 Following the permutation, a diffusion operation is performed to obtain the ciphertext image C. The algorithm for this step is detailed as follows:

$$\begin{cases} C(i) = C(i-1) \times S_2(i) \times P_1(i), \\ C(i) = C(i+1) \times S_2(i) \times P_1(i). \end{cases} \tag{12}$$

Step 5 Both the permutation and diffusion steps can be repeated multiple times to achieve the final ciphertext image C.

A flowchart of the encryption process is illustrated in Fig. 9. The decryption steps are the inverse of the aforementioned encryption operations.

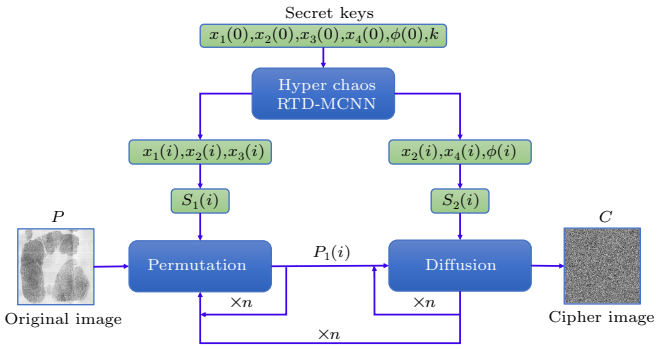


Fig. 9. Flow chart of the proposed image encryption algorithm.

4.2. Random sequence generation and randomness testing

Before performing image encryption, it is essential to assess the randomness of the chaotic sequences. Currently, randomness testing for chaotic image encryption sequences primarily relies on the FIPS 140-2 and NIST SP800-22 standards. This study selects the latter, which has more stringent requirements, as the criterion for evaluating the randomness of the chaotic sequences generated by the RTD-MCNN. The NIST SP800-22 standard includes 15 testing methods to assess the randomness of sequences. Each method produces a P-value between [0,1]: a P-value greater than the predetermined threshold indicates a passing test. In this study, the software's default threshold of 0.01 is used.

In this subsection, the initial parameters for Eq. (5) are set as $k = 0.3$ and ICs $= [0.1, 0.0, 0.1, 0.1, 0.1]$, with a simulation time of $t = 50000$. The resulting five sets of floating-point values $x(: 1000 : 7e7)$ are grouped according to the rules of Eq. (9), then converted to binary encoding and stored in a text file. This text file is subsequently processed using the official NIST testing software, yielding results presented in Table 3, where P_1-value and P_2-value correspond to the test results for random sequences S_1 and S_2, respectively. The results indicate that the chaotic sequences generated by RTD-MCNN under the current conditions meet all the testing standards of NIST SP800-22, demonstrating excellent randomness and thereby achieving satisfactory encryption performance.

Table 3. Results of the NIST800-22 test for pseudo-random sequences S1 and S2.

Test name	P_1-value	Result	P_2-value	Result
Frequency	0.867692	pass	0.085587	pass
Block frequency	0.719747	pass	0.419021	pass
Cumulative sums	0.014550	pass	0.574903	pass
Runs	0.275709	pass	0.350485	pass
Longest run	0.867692	pass	0.289667	pass
Rank	0.574903	pass	0.964295	pass
FFT	0.816537	pass	0.739918	pass
Non-overlapping template	0.026948	pass	0.025193	pass
Overlapping template	0.162606	pass	0.145326	pass
Universal	0.366918	pass	0.983453	pass
Approximate entropy	0.514124	pass	0.798139	pass
Random excursions	0.062821	pass	0.110952	pass
Random excursions variant	0.145326	pass	0.054199	pass
Serial	0.075719	pass	0.334538	pass
Linear complexity	0.213309	pass	0.759756	pass

4.3. Encryption results and security performance analysis

This section provides a comprehensive security analysis of the encryption results, focusing on histogram analysis, correlation analysis, information entropy, pixel number change rate (NPCR), uniform average change intensity (UACI), key sensitivity and robustness. The encryption outcomes will be illustrated throughout the analysis. In this section, three images — a palm print, a fingerprint and a footprint — commonly encountered in case processing are selected as the initial images for encryption, all with a resolution of 256×256 pixels.

4.3.1. Histogram analysis

The histogram reflects the frequency of each grayscale value present in an image. Different initial images exhibit distinct histogram distributions; generally, unencrypted images show uneven pixel value distributions. An effective encryption algorithm should produce a histogram distribution that is uniform, preventing attackers from extracting useful information from it. As shown in Fig. 10, the initial images — palm print, fingerprint and footprint — are represented in Figs. 10(a1)–10(a3), while their corresponding ciphertext images are displayed in Figs. 10(c1)–10(c3). Visually, it is evident that the ciphertext images have become unrecognizable. The histograms for the original and ciphertext images are shown in Figs. 10(b1)–10(b3) and 10(d1)–10(d3), respectively. The results indicate that the histograms of the initial images are unevenly distributed across various pixel intensities, whereas the histograms of the ciphertext images are approximately uniform. These results demonstrate that the encryption algorithm operates successfully, mitigating the risk of certain statistical attacks.

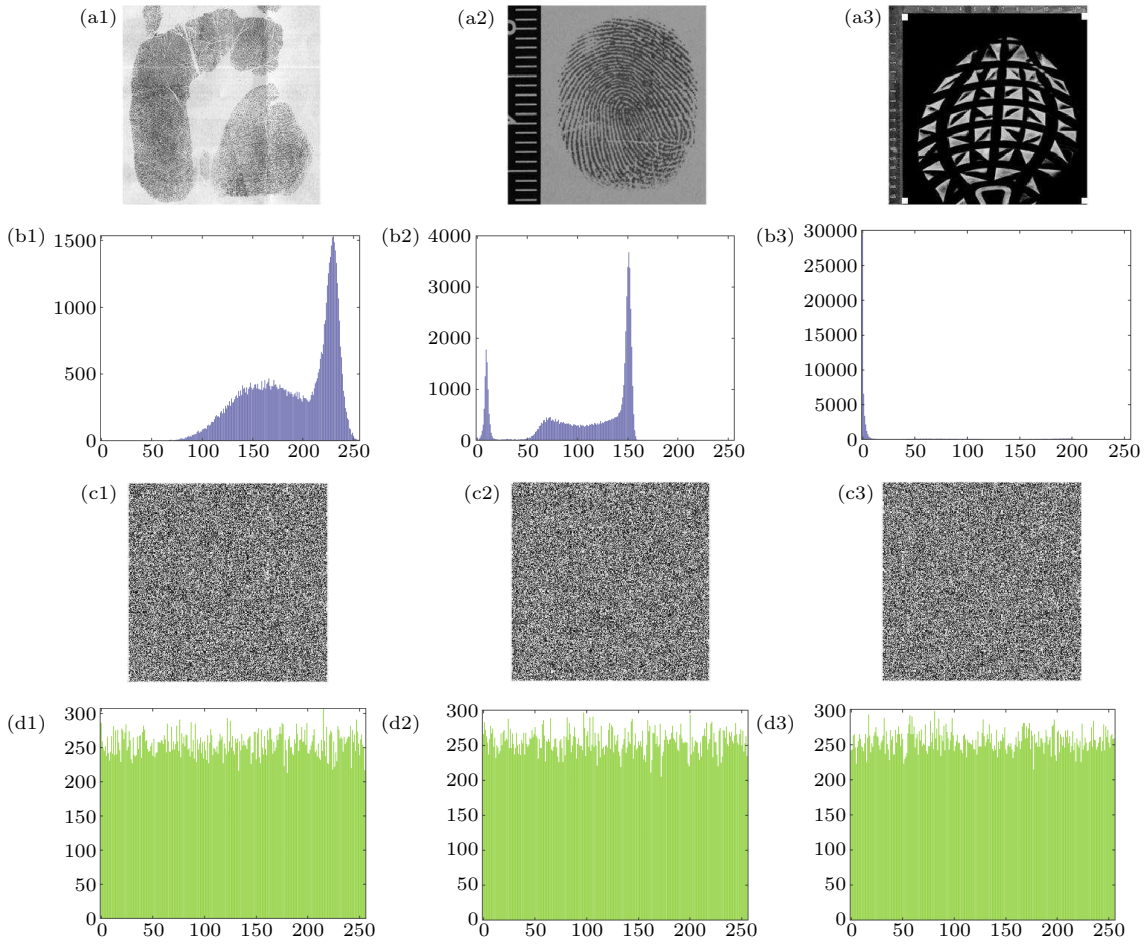


Fig. 10. The results of the histogram analysis: (a1)–(a3) original images; (b1)–(b3) histograms of the original images (a1)–(a3); (c1)–(c3) ciphertext images; (d1)–(d3) histograms of the ciphertext images (c1)–(c3).

4.3.2. Correlation analysis

In criminal cases, the details of subjects in forensic photographs often exhibit very high correlation. The correlation between image pixels is frequently a critical factor that attackers exploit to break encryption algorithms. Therefore, aiming to thoroughly eliminate correlation within images should be a primary focus for enhancing encryption algorithms. In this study, the statistical analysis of correlation is presented as follows:

$$\begin{cases} R_{xy} = \dfrac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \\ D(x) = (1/N)\sum_{i=1}^{N}(x_i - E(x))^2, \\ \operatorname{cov}(x,y) = (1/N)\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)), \\ E(x) = (1/N)\sum_{i=1}^{N} x_i, \end{cases} \tag{13}$$

where x and y represent any two adjacent pixel values, N is the number of randomly selected pixel pairs in the image, R_{xy} is the correlation coefficient of the image and $\operatorname{cov}(x,y)$ denotes the covariance between adjacent pixels. $E(\cdot)$ and $D(\cdot)$ represent the expectation and variance, respectively. In this study, 5000 pixels were selected from the image, and the correlation coefficients were calculated in the vertical, horizontal, diagonal and anti-diagonal directions, as shown in Fig. 11, with the corresponding values recorded in Table 4. The results indicate that, prior to encryption, the correlation coefficients of the plaintext images in all directions exceed 0.8, indicating strong correlation, which is visually represented by a pronounced linear relationship in Figs. 11(a1)–11(a3). In contrast, the correlation coefficients of the encrypted images in all directions approach 0, demonstrating very low interpixel correlation. This low correlation is illustrated in Figs. 11(b1)–11(b3), where the pixels are uniformly distributed. Therefore, the proposed encryption scheme exhibits strong robustness.

Table 4. Correlation coefficient of different images before and after encryption.

Images	Correlation coefficient				
	Ciphertext/plaintext	Vertical	Horizontal	Diagonal	Anti-diagonal
Palm print	plaintext	0.8279	0.8383	0.8003	0.8067
	ciphertext	0.0103	0.0149	−0.0167	0.0027
Fingerprint	plaintext	0.9115	0.9341	0.8771	0.8562
	ciphertext	−0.0131	0.0094	0.0057	−0.0029
Footprint	plaintext	0.8918	0.9312	0.8357	0.8588
	ciphertext	−0.0103	−0.0016	0.0054	−0.0025

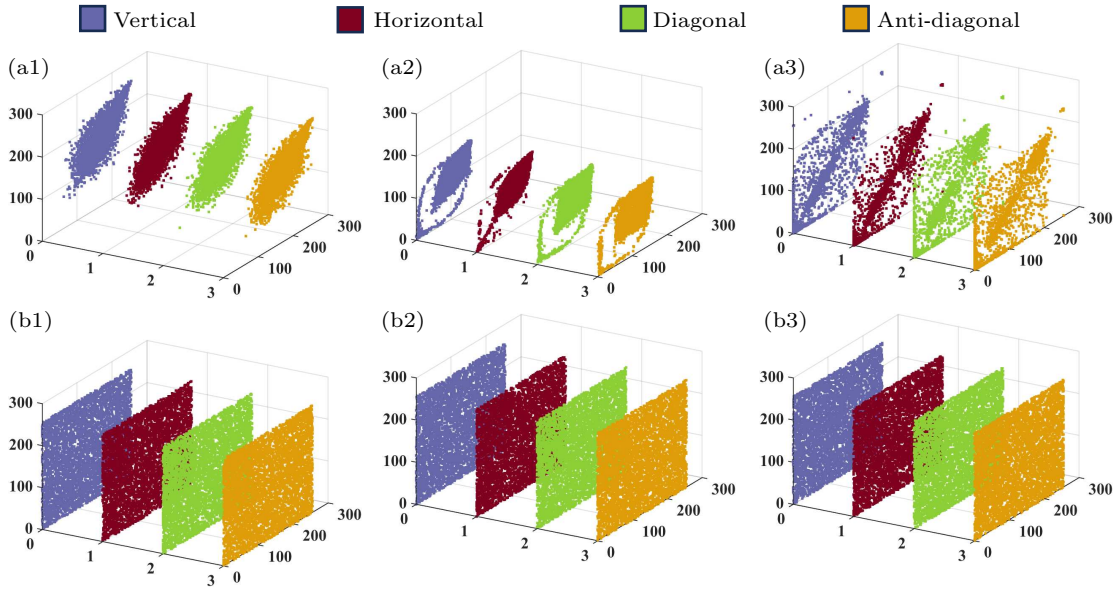


Fig. 11. Correlation of different images: (a1) correlation of the palm print image; (a2) correlation of the fingerprint image; (a3) correlation of the footprint image; (b1)–(b3) correlation of the corresponding ciphertext images.

4.3.3. Entropy

The statistical properties of digital image information can be reflected through information entropy. According to Shannon's theorem, for a digital image with a maximum pixel grayscale value of 256, the ideal information entropy should be 8. The formula for calculating information entropy is as follows:

$$H(x) = -\sum_{i=0}^{2^N-1} p(x_i)\log_2[p(x_i)], \tag{14}$$

where H is the information entropy, $p(x_i)$ is the probability of occurrence of the pixel value x_i in the image and N is the bit depth of the image P. Based on the calculations from Eq. (14), the information entropy for the images before and after encryption is presented in Table 5. The information entropy of the encrypted images shows a significant increase and approaches the ideal value of 8. Therefore, the proposed encryption scheme demonstrates a strong ability to resist statistical attacks.

Table 5. Information entropy, NPCR and UACI before and after image encryption.

Images	Ciphertext/plaintext	Entropy	NPCR	UACI
Palm print	plaintext	7.0096	99.6102	33.4534
	ciphertext	7.9969		
Fingerprint	plaintext	6.3816	99.6087	33.4886
	ciphertext	7.9971		
Footprint	plaintext	4.4887	99.6068	33.4516
	ciphertext	7.9977		

4.3.4. Differential attack analysis

There is a specific type of attack on image encryption known as differential attack, where an attacker alters a single pixel in the plaintext image and then compares the encrypted images before and after the alteration. By analyzing the differences between the two encrypted images, the attacker may deduce the encryption pattern. The metrics NPCR and UACI quantitatively measure the ability of the encryption algorithm to resist differential attacks. In brief, the ideal values for NPCR and UACI are 99.6094% and 33.4635%, respectively. The closer the two metrics for the encrypted images are to these ideal values, the stronger the resistance of the encryption algorithm to differential attacks. The formulae for calculating NPCR and UACI for images of size $M \times N$ can be expressed as follows:

$$\begin{cases} \mathrm{NPCR}(C_1,C_2) = \sum_{i=1}^{M}\sum_{j=1}^{M} \frac{D(i,j)}{M\times N} \times 100\%, \\ \mathrm{UACI}(C_1,C_2) = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{M} \frac{|C_1(i,j)-C_2(i,j)|}{255} \times 100\%, \end{cases} \tag{15}$$

where

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_1(i,j), \\ 1, & \text{if } C_1(i,j) \neq C_1(i,j). \end{cases} \tag{16}$$

The NPCR and UACI values for various digital images calculated using Eqs. (15) and (16) are listed in Table 5, showing values that are very close to the ideal values. This indicates that the encryption scheme based on RTD-MCNN demonstrates strong resistance to differential interference attacks.

4.3.5. Key sensitivity

Given the chaotic characteristics of RTD-MCNN, the chaotic sequences it generates are extremely sensitive to the initial values in the key. For the initial values $(x_1(0), x_2(0), x_3(0), x_4(0), \phi(0))$, this study selects $k = 0.3$ along with $(0.1, 0, 0.1, 0.1, 0.1)$ for the encryption algorithm and $(0.1+10^{-17}, 0, 0.1, 0.1, 0.1)$ for the decryption algorithm. From this, the decrypted images obtained by using the correct key and the wrong key are shown in Fig. 12. The results indicate that even with an error magnitude of the order of 10^{-17},

the incorrect key remains incapable of decrypting the original image information, demonstrating that the proposed encryption algorithm possesses excellent key sensitivity.

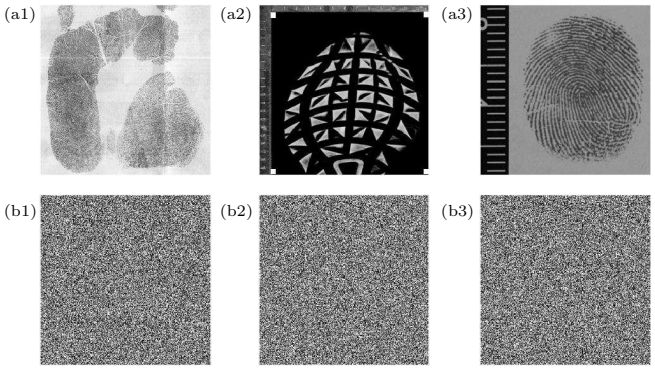


Fig. 12. The results of the key sensitivity analysis. (a1)–(a3) The decrypted images with the correct key $(0.1,0,0.1,0.1,0.1)$. (b1)–(b3) The decrypted images with an incorrect key $(0.1+10^{-17},0,0.1,0.1,0.1)$.

4.3.6. Robustness

The ability to withstand data loss and noise attacks reflects the robustness of the encryption algorithm. In this study, the ciphertext images had 2% and 8% of their data obscured, as shown in Figs. 13(a1) and 13(a2), with the corresponding decrypted images displayed in Figs. 13(a3) and 13(a4). Clearly, even with portions of the ciphertext cropped, the decrypted images still retain recognizable information about the original images. Additionally, before decryption, varying levels of salt-and-pepper noise, namely, 0.5%, 1%, 2% and 5%, were added to the ciphertext images, with the decrypted results shown in Figs. 13(b1)–13(b4). The results indicate that after introducing 5% salt-and-pepper noise, although some pixel values in the decrypted images changed compared with the original images, the overall essential information in the original images remains discernible. Therefore, the proposed encryption scheme demonstrates strong robustness.

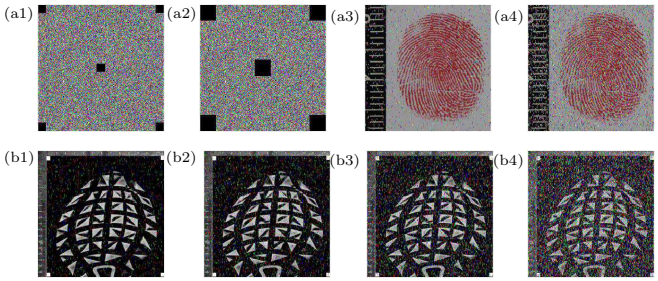


Fig. 13. The results of the robustness analysis: (a1) ciphertext image with 2% data loss; (a2) ciphertext image with 8% data loss; (a3) decrypted image corresponding to (a1); (a4) decrypted image corresponding to (a2). (b1)–(b4) Decrypted images corresponding to the addition of 0.5%, 1%, 2%, and 5% salt-and-pepper noise, respectively.

Table 6. Comparison of encryption effects under different encryption schemes.

Paper	Image type	Entropy	NPCR/UCAI	Key sensitivity	Robustness	Time (s)
[74]	industrial images 256×256	7.9973	99.6048 33.4728	10^{-13}	–	–
[75]	ordinary images 256×256	7.9971	99.6566 33.4647	10^{-15}	yes	–
[5]	ordinary images –	–	99.8700 33.2200	10^{-15}	–	0.53
[76]	Lena's images 256×256	7.9978	– –	–	yes	–
This paper	evidence images 256×256	7.9977	99.6068 33.4516	10^{-17}	yes	0.35

4.4. Comparison of encryption performance

With the rapid development of the information age, traditional encryption schemes such as DES, IDEA and RSA have lost their advantages in handling massive images and the high correlation of image data. These traditional algorithms often exhibit low computational efficiency and insufficient security when faced with large-scale data. Consequently, researchers have begun to explore new encryption methods to meet the demands of modern information security. Image encryption schemes based on memristive neural networks have gained widespread attention due to their lightweight nature and high reliability. These schemes leverage the properties of memristors to significantly reduce computational and storage complexity while maintaining encryption strength. The traditional MHNN encryption algorithm involves division operations and a hyperbolic tangent function based on exponential operations, resulting in lower computational efficiency due to floating-point calculations during implementation. In contrast, the lightweight design of the RTD-MCNN-based scheme incorporates only multiplication, addition and negation operations, allowing for direct results through bitwise operations. This approach demonstrates significant advantages in processing speed and resource consumption. Table 6 presents a comparison of the encryption effectiveness between the proposed scheme and recent encryption methods. It is evident that the RTD-MCNN-based encryption algorithm exhibits superior key sensitivity, along with exceptional performance across various other metrics. The encryption scheme in this paper utilizes a computer CPU with an 11th generation Intel Core i5-1155G7@2.5 GHz, with the software platform being MAT-

LAB 2021b; the encryption time is 0.38 s. In comparison, reference [5] containing the hyperbolic tangent function uses an AMD Ryzen 7 5800H CPU with an encryption time of 0.58 s. Therefore, under conditions of lower processor performance, the scheme presented in this paper has an advantage in encryption time.

5. Conclusion

This paper proposes a memristor-based neural network model, RTD-MCNN, which employs a memristor model with an absolute value function. In analyzing the dynamical behavior of RTD-MCNN, this study conducts an in-depth exploration of various aspects, including equilibrium points, phase diagrams, time series, Lyapunov exponents, bifurcation diagrams and attractor basins, revealing that the model is a multistable and ultra-chaotic system influenced by the internal parameters of the memristor.

Based on the dynamical behavior of the RTD-MCNN, this paper designs a simple yet highly reliable encryption scheme intended for the protection of digital images in police forensic applications. The two sets of chaotic sequences used in the encryption process have passed the NIST SP800-22 tests, indicating that they fully meet high encryption standards. By analyzing the histograms, correlation coefficients, information entropy, key sensitivity and indicators such as NPCR and UACI before and after encryption, the final effectiveness and security of the encryption scheme are validated.

Acknowledgments

Project supported by the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 24A0248), the National Key Research and Development Program "National Quality Infrastructure System" Special Project (Grant No. 2024YFF0617900), and the Hefei Minglong Electronic Technology Co., Ltd. (Grant Nos. 2024ZKHX293, 2024ZKHX294, and 2024ZKHX295).

References

[1] Saadati M, Khodaei S S and Jamali Y 2024 *Communications in Nonlinear Science and Numerical Simulation* **137** 108133
[2] Mou J, Cao H, Zhou N and Cao Y 2024 *IEEE Transactions on Cybernetics* **54** 7333
[3] Zhu J, Jin J, Chen C, Wu L, Lu M and Ouyang A 2024 *IEEE Transactions on Emerging Topics in Computational Intelligence* 1
[4] Xu Q, Wang K, Chen M, Parastesh F and Wang N 2024 *Chaos Soliton. Fract.* **181** 114654
[5] Deng Q, Wang C, Sun Y, Deng Z and Yang G 2024 *IEEE Transactions on Circuits and Systems I: Regular Papers* 1–12
[6] Hodgkin A L and Huxley A F 1952 *Journal of Physiology* **117** 500
[7] Hindmarsh J L and Rose R 1984 *Proceedings of the Royal Society of London. Series B. Biological sciences* **221** 87
[8] Morris C and Lecar H 1981 *Biophysical Journal* **35** 193
[9] Chua L and Yang L 1988 *IEEE Transactions on Circuits and Systems* **35** 1257
[10] Hopfield J J 1982 *Proc. Natl. Acad. Sci. USA* **79** 2554
[11] Liao Z, Ouyang J and Ma M 2024 *Nonlinear Dynam.* 1
[12] Yu F, Wu C, Xu S, Yao W, Xu C, Cai S and Wang C 2025 *Signal, Image and Video Processing* **19** 77
[13] Zhang S, Yao W, Xiong L, Wang Y, Tang L, Zhang X and Yu F 2024 *Chaos Soliton. Fract.* **187** 115408
[14] Deng Z, Wang C, Lin H, Deng Q and Sun Y 2024 *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 1
[15] Dong Z, Duan S, Hu X, Wang L and Li H 2014 *Scientific World Journal*
[16] Ji X, Dong Z, Han Y, Lai C S and Qi D 2023 *IEEE Transactions on Circuits and Systems for Video Technology* **33** 7928
[17] Ji X, Dong Z, Zhou G, Lai C S and Qi D 2024 *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **54** 5137
[18] Itoh M and Chua L O 2009 *Int. J. Bifurc. Chaos* **19** 3605
[19] Itoh M and Chua L O 2008 *Int. J. Bifurc. Chaos* **18** 3183
[20] Itoh M, Julián P and Chua L O 2001 *Int. J. Bifurc. Chaos* **11** 2913
[21] Chua L 1971 *IEEE Transactions on Circuit Theory* **18** 507
[22] Wan Q, Yang Q, Liu T, Chen C and Shen K 2024 *Chaos Soliton. Fract.* **189** 115584
[23] Yu F, Xu S, Lin Y, Gracia Y M, Yao W and Cai S 2024 *Int. J. Bifurc. Chaos* **34** 2450068
[24] Kong X, Yu F, Yao W, Cai S, Zhang J and Lin H 2024 *Neural Networks* **171** 85
[25] Yu F, Kong X, Yao W, Zhang J, Cai S, Lin H and Jin J 2024 *Chaos Soliton. Fract.* **179** 114440
[26] Wan Q, Liu J, Liu T, Sun K and Qin P 2024 *Neural Networks* **174** 106268
[27] Yao W, Fang J, Yu F, Xiong L, Tang L, Zhang J and Sun Y 2024 *Chaos* **34** 073149
[28] Zhang S, Chen C, Zhang Y, Cai J, Wang X and Zeng Z 2024 *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 1
[29] Yuan Y, Yu F, Tan B, Huang Y, Yao W, Cai S and Lin H 2025 *Chaos* **35** 013121
[30] Tang D, Wang C, Lin H and Yu F 2024 *Nonlinear Dynam.* **112** 1511
[31] Luo D, Wang C, Deng Q and Sun Y 2024 *Nonlinear Dynam.* 1
[32] Zhang S, Li Y, Lu D and Li C 2024 *Chaos Soliton. Fract.* **184** 115056
[33] Xu Q, Wang K, Feng C, Fan W and Wang N 2024 *Chaos Soliton. Fract.* **189** 115646
[34] Li J, Wang C and Deng Q 2024 *Nonlinear Dynam.* **112** 14463
[35] Lin H, Deng X, Yu F and Sun Y 2024 *IEEE Internet of Things Journal* **11** 29878
[36] Wang C, Luo D, Deng Q and Yang G 2024 *Chaos Soliton. Fract.* **187** 115471
[37] Qi A X, Zhu B D and Wang G Y 2022 *Chin. Phys. B* **31** 020502
[38] Wu H, Bian Y, Zhang Y, Guo Y, Xu Q and Chen M 2023 *Chaos Soliton. Fract.* **177** 114201
[39] Chen Q, Li B, Yin W, Jiang X and Chen X 2023 *Chaos Soliton. Fract.* **171** 113440
[40] Aguirre F, Sebastian A, Le Gallo M, Song W, Wang T, Yang J J, Lu W, Chang M F, Ielmini D, Yang Y, *et al.* 2024 *Nat. Commun.* **15** 1974
[41] Zhou L, Zhang H, Tan F and Liu K 2024 *Transactions of the Institute of Measurement and Control* **46** 2035
[42] Sun J, Li C, Wang Z and Wang Y 2024 *IEEE Transactions on Industrial Informatics* **20** 3778
[43] Liu X, Mou J, Zhang Y and Cao Y 2023 *IEEE Transactions on Industrial Electronics* **71** 5094
[44] Yang F, Ma J and Wu F 2024 *Chaos Soliton. Fract.* **187** 115361
[45] Kong X, Yu F, Yao W, Xu C, Zhang J, Cai S and Wang C 2024 *Applied Mathematical Modelling* **125** 351
[46] Zhou L, Lin H and Tan F 2024 *Neurocomputing* **577** 127384
[47] Yao W, Liu J, Sun Y, Zhang J, Yu F, Cui L and Lin H 2024 *Nonlinear Dynam.* **112** 693
[48] Ma M, Xiong K, Li Z and He S 2024 *Chin. Phys. B* **33** 028706
[49] Feng W, Wang Q, Liu H, Ren Y, Zhang J, Zhang S, Qian K and Wen H 2023 *Fractal and Fractional* **7** 887
[50] Yu F, Yuan Y, Wu C, Yao W, Xu C, Cai S and Wang C 2024 *Nonlinear Dynam.* **112** 2331
[51] Wang M J and Gu L 2024 *Chin. Phys. B* **33** 020504
[52] Yu F, Shen H, Yu Q, Kong X, Sharma P K and Cai S 2023 *IEEE Transactions on Network Science and Engineering* **10** 845
[53] Lin H, Deng X, Yu F and Sun Y 2024 *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*
[54] Wang C, Tang D, Lin H, Yu F and Sun Y 2024 *Expert Systems with Applications* **242** 122513
[55] Chen F J and Li J B 2008 *Int. J. Bifurc. Chaos* **18** 3439

[56] Huang X, Zhao Z, Wang Z and Li Y 2012 *Neurocomputing* **94** 13
[57] Jiao X, Yuan M, Tao J, Sun H, Sun Q and Chen Z 2023 *Chin. Phys. B* **32** 010507
[58] Sun Q, He S, Sun K and Wang H 2022 *Chin. Phys. B* **31** 120501
[59] Jin J, Lei X, Chen C and Li Z 2024 *Mathematics and Computers in Simulation*
[60] Yu F, Zhang W, Xiao X, Yao W, Cai S, Zhang J, Wang C and Li Y 2024 *Fractal and Fractional* **8** 271
[61] Feng W, Yang J, Zhao X, Qin Z, Zhang J, Zhu Z, Wen H and Qian K 2024 *Mathematics* **12** 3917
[62] Yu F, Xu S, Lin Y, He T, Wu C and Lin H 2024 *Mathematics* **12** 2227
[63] Chua L 2014 *Semiconductor Science and Technology* **29** 104001
[64] Deng W and Ma M 2024 *Chin. J. Phys.* **91** 966
[65] Itoh M 2019 arXiv:1902.05167 [cs.NE]
[66] Hu X, Feng G, Duan S and Liu L 2015 *Neurocomputing* **162** 150
[67] Lai Q, Yang L, Hu G, Guan Z H and Iu H H C 2024 *IEEE Transactions on Cybernetics* **54** 4039
[68] Wen H and Lin Y 2023 *Journal of King Saud University-Computer and Information Sciences* **35** 101612
[69] Ma X, Wang Z and Wang C 2024 *Int. J. Bifurc. Chaos* **34** 2450170
[70] Zhou L, Lin Z and Tan F 2024 *Physica Scripta* **99** 045247
[71] Wen H and Lin Y 2024 *Expert Systems with Applications* **237** 121514
[72] Sun J, Li C, Wang Z and Wang Y 2024 *IEEE Transactions on Industrial Informatics* **20** 3778
[73] Feng W, Zhang J, Chen Y, Qin Z, Zhang Y, Ahmad M and Wozniak M 2024 *Expert Systems with Applications* **246** 123190
[74] Yu F, Lin Y, Yao W, Cai S, Lin H and Li Y 2025 *Neural Networks* **182** 106904
[75] Singh A K, Chatterjee K and Singh A 2023 *IEEE Transactions on Industrial Informatics* **19** 1957
[76] Lai Q, Wan Z and Kuate P D K 2023 *IEEE Transactions on Circuits and Systems I: Regular Papers* **70** 1324