Multimedia Tools and Applications (2022) 81:7365–7391
https://doi.org/10.1007/s11042-021-11803-1

An image encryption scheme based on chaotic logarithmic map and key generation using deep CNN

Uğur Erkan[1] · **Abdurrahim Toktas**[2] · **Serdar Enginoğlu**[3] · **Enver Akbacak**[4] · **Dang N. H. Thanh**[5]

Received: 7 June 2021 / Revised: 6 September 2021 / Accepted: 14 December 2021 /
Published online: 26 January 2022

Abstract

A secure and reliable image encryption scheme is presented, which depends on a novel chaotic log-map, deep convolution neural network (CNN) model, and bit reversion operation for the manipulation process. CNN is utilized to generate a public key to be based on the image in order to enhance the key sensitivity of the scheme. Initial values and control parameters are then obtained from the key to be used in the chaotic log-map, and thus a chaotic sequence is produced for the encrypting operations. The scheme then encrypts the images by scrambling and manipulating the pixels of images through four operations: permutation, DNA encoding, diffusion, and bit reversion. The encryption scheme is precisely examined for the well-known images in terms of various cryptanalyses such as key-space, key sensitivity, information entropy, histogram, correlation, differential attack, noisy attack, and cropping attack. To corroborate the image encryption scheme, the visual and numerical results are even compared with available scores of the state of the art. Therefore, the proposed log-map-based image encryption scheme is successfully verified and validated by superior absolute and comparative results. As future work, the proposed log-map can be extended to combinational multi-dimensional with existing efficient chaotic maps.

Keywords Image encryption · Chaotic map · Logarithmic map · Deep convolution neural network (CNN) · Bit reversion

✉ Uğur Erkan
ugurerkan@kmu.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Karamanoğlu Mehmetbey University, 70200 Karaman, Turkey

[2] Department of Electrical-Electronic Engineering, Faculty of Engineering, Karamanoğlu Mehmetbey University, 70200 Karaman, Turkey

[3] Department of Mathematics, Faculty of Arts and Sciences, Çanakkale Onsekiz Mart University, Çanakkale, Turkey

[4] Department of Computer Engineering, Faculty of Engineering, Haliç University, Sütlüce Mah. Imrahor Cad. No: 82 Beyoglu, 34445 Istanbul, Turkey

[5] Department of Information Technology, School of Business Information Technology, University of Economics Ho Chi Minh City, Ho Chi Minh City, Vietnam

1 Introduction

In line with progress in computer and network technologies, real-time messaging and data transferring have become inevitable, particularly for medical, military, and education applications [3, 7, 61, 19]. In general, the data is transmitted through a wide area network (WAN), likely exposed to cyber threats such as network attacks, denial-of-service, man-in-the-middle, and phishing [65]. Therefore, it should be transferred after encrypting the data for providing information security, making the data encryption techniques the most crucial task [14, 34]. The well-known encryption techniques that especially emerged for text encryption are data encryption standard (DES), triple-DES (3DES), international data encryption algorithm (IDEA), and an advanced encryption standard (AES). Steganography has become a crucial tool for preventing sensitive communications in various applications [1, 2]. In case they are implemented to the image encryption, the security is low due to high correlation and a large amount of data [27, 49].

Image encryption techniques based on the spatial domains such as Deoxyribonucleic Acid (DNA) [11, 12, 14, 18, 20, 47, 57, 59, 61] coding, chaos [5, 6, 11, 12, 15, 18, 20, 22, 26, 33, 36, 47, 55, 59, 63] and cellular automata [18], and the frequency domains like Fourier [20] and wavelet transform [29] have been widely exploited for the last decades [57, 66]. There have been different image encryption studies based on machine learning [16, 23, 25, 32, 38, 40, 42, 44, 45, 67]. Chaos-based schemes are the most employed techniques due to high randomization, complexity, sensitivity to initial conditions, and system parameters. Chaos-based image encryption schemes are processed through two stages, in general. In the first stage, the main key is constituted with XOR operation between a secret key and a public key. The image is then encrypted in the second stage by using the main key. It is crucial to use an encryption scheme sensitive to the main key to achieve high secure and reliable ciphertext image. Therefore, a key derived from the plaintext image is a quite reasonable and effective way. In the second stage, the image's pixels are interchanged in position and manipulated in color tonal intensity using permutation and diffusion operations, respectively. The permutation and diffusion operations are carried out using chaotic maps with various dimensions and DNA coding managed by the main key. The operations are conducted by a sequence produced by the chaotic map utilizing the initial value and control parameter. Therefore, chaotic maps play a decisive role in achieving success in an encryption scheme. The performance of encryption schemes is evaluated using different cryptanalyses such as key-space, key sensitivity, information entropy, histogram, correlation, differential attack, noisy attack, and cropping attack [13]. Many image encryption schemes have been proposed in the literature [5, 6, 11, 12, 15, 18, 20, 22, 26, 33, 36, 47, 55, 59, 63]. They are mainly based on various chaotic maps such as a logistic-adjusted-sine map [12], a modified logistic map [22], a sine powered chaotic map [36], a polynomial chaotic map [5], a cosine-transform-based chaotic map [26], integrating three basic chaotic maps [33], sine and logistic map [6], sine-sine map [55], four-wing chaotic system [11], Henon and sine maps [59]. Fourier transform and Lorenz map [20], a combination of sine map, Chebyshev map and a linear function [15], memristive chaotic system [63], cellular automata [18], coupled map lattice [47]. When those suggested schemes are surveyed in the view of encryption performance, it is seen that they have their own strong and weak capabilities across the encrypting operations give successful results for particular cryptanalyses which were performed. Hence, a more secure image encryption method is necessary to improve security for all cryptanalyses.

In this study, an image encryption scheme based on a novel chaotic map and key generation is proposed. The scheme is built on a logarithmic chaotic map and public key generated through a deep convolution neural network (CNN). The new contributions of the presented study can be emphasized as follows:

- A novel chaotic map referred to as log-map is introduced for producing chaotic sequences.
- A sensitive key generation method through a designed CNN model based on VGG16 architecture is proposed for feature extraction, which depends on the image to be encrypted.
- An operation so-called bit reversion is suggested for the manipulation.
- The outperforming results are compared with the state of the arts.

First, a main public key is obtained with XOR operation between a secret key and the public key generated through the deep CNN model. In this way, a very sensitive key is generated depending on the image. Then, four initial values and four control parameters are produced to be used in the chaotic log-maps, a part of the encrypting operations such as permutation, DNA encoding, diffusion, and bit reversion, respectively. Afterward, the performance of the encryption scheme for the well-known images is comparatively analyzed across the reliable metrics regarding visual and numerical views. Based on observed results, the proposed scheme outperforms state-of-the-art techniques thanks to having a sensitive key, chaotic log-map and impactful bit reversion operation.

2 Related studies

Several image encryption schemes have been proposed in the literature [5, 6, 11, 12, 15, 18, 20, 22, 26, 33, 36, 47, 55, 59, 63]. An image encryption scheme was suggested in [12], a 2D logistic-adjusted-sine map for permutation, and a 1D chaotic map was introduced for diffusion. The authors in [22] proposed a scheme based on the integer-based key generation with a modified logistic map. In [36], a 1D sine powered chaotic map was developed for image encryption. In [5], a polynomial 1D chaotic map was constituted by combining few chaotic maps. A cosine-transform-based chaotic map was introduced in [26]. In [33], a chaotic map was obtained by integrating three basic 1D maps conducting cascade, nonlinear combination, and switch operations. In [6], a method based on YoloV3 object detection and chaotic image map was developed. The authors in [55] defined a scheme combining two sine maps in the permutation stage and utilizing key-streams produced by the sine-sine-map. In [11], an image cryptosystem based on dynamic DNA encryption and chaos was suggested. [59] introduced a 2D chaotic map generated by connecting Henon map and sine map. A method for image encryption using fractional Fourier transform, DNA sequence operation, and chaos theory was developed in [20] . The authors in [15] put forward a cryptosystem based on a 2D chaotic map derived from a sine map, Chebyshev map, and a linear function. In [63], a 4D dimensional memristive chaotic system was constructed based on Liu's chaotic system by introducing a flux-controlled memristor model. In [18], a multiple-image encryption scheme based on DNA sequence and image matrix was studied to provide fast and secure indexes. The authors in [47] implemented a Pareto-optimal image encryption scheme using a coupled map lattice chaos function and DNA combination. In case those suggested schemes are surveyed in the view of encryption performance, although they have their own strong and weak abilities, they give successful results for particular cryptanalyses performed. In other words, they are not secure against overall cyber-attacks. For example, the key-space results in [15, 47, 55, 63] comparatively seem the lowest within the analysis carried out in [2, 12, 15, 18, 20, 27, 47, 49, 55, 57, 59]. The information entropies can be assessed as better [12, 18, 22, 59, 63], moderate [11, 15, 20] and worse [5, 6, 47] in comparison among each other. While the correlation results in [5, 11, 18, 22, 26, 36, 55, 59] are lower, those in [6, 12, 21, 33, 63] are moderate, and ones in [20, 47] are higher. The resistances of the schemes counter the differential attacks can be sorted from the strongest to the weakest as [11, 15, 22], [33, 36, 59, 63], and [5, 6, 12, 18, 20, 26, 47, 55]. The scheme in [55] appears better against cropping and noise attacks among the available results [5, 6, 11, 15, 26, 33, 36, 55, 59, 63]. They are elaborately compared with the proposed scheme in Section 5. It is evident that the visual and numerical results of the log-map-based image encryption are better than those of the suggested studies.

3 Overview of the proposed image encryption scheme based on log-map

In the proposed image encryption scheme whose block diagram is given in Fig. 1, a new chaotic map so-called log-map, a feature extraction-based public key, and bit reversion is introduced through the scheme. First, the public key is generated using CNN for feature extraction depending on the image so as to improve the key sensitivity. As the secret key is available at both sender and receiver sides, the public key is sent together with the ciphertext image. A sensitive main key is obtained by XOR operation between the public key and secret key. Afterward, the ciphertext image is achieved through the four operation stages: permutation, DNA coding, diffusion, and bit reversion. The chaotic sequence is produced using the log-map by serving the main key as the control parameter and initial value to the map. The four operation stages are then processed according to the chaotic sequence and the image is encoded with the DNA rules.

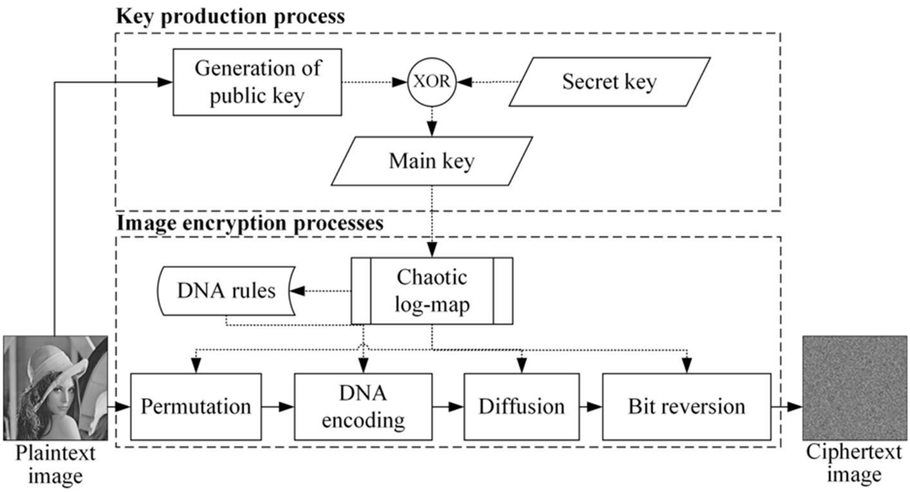

Fig. 1 An overview of the proposed image encryption scheme

3.1 Generation of the public key from the image based on feature extraction using deep CNN

This section describes the generation of a public key via a deep CNN model depicted in Fig. 2 by employing VGG16 architecture [41] with the pre-trained network having an Image-Net dataset with 1000 categories. Besides VGG16, there are various pre-trained architectures for CNN. The well-known architectures are AlextNet [31], Inception-Net [48], ResNet [24], DenseNet [28], EfficientNet [50] and Big Transfer (BiT) [30] and their variants. Each pre-trained architecture has a higher classification accuracy score than a previous version. AlexNet has 60 M parameters and 3 convolutional layers. VGG16 with 138 M parameters has 13 convolutional layers and uses a smaller size of filters than AlexNet. VGG19 is an improved version of VGG16 in terms oflayers. In InceptionNet, also known as InceptionV1 or GoogleNet, modules are stacked. Modules consist of parallel convolutional layers having different kernels. It consists of 9 stacked inception modules having 5 M parameters. ResNet addresses the saturation of deeper networks originated due to vanishing gradients during backpropagation by skip the connections. The input of a layer is directly connected to the output of that layer with a skip connection. In DenseNet, there are dense blocks between convolution layers. In dense blocks, each layer has extra inputs from previous layers. Unlike ResNet, feature maps are combined via concatenation. EffientNet is developed based on proposing

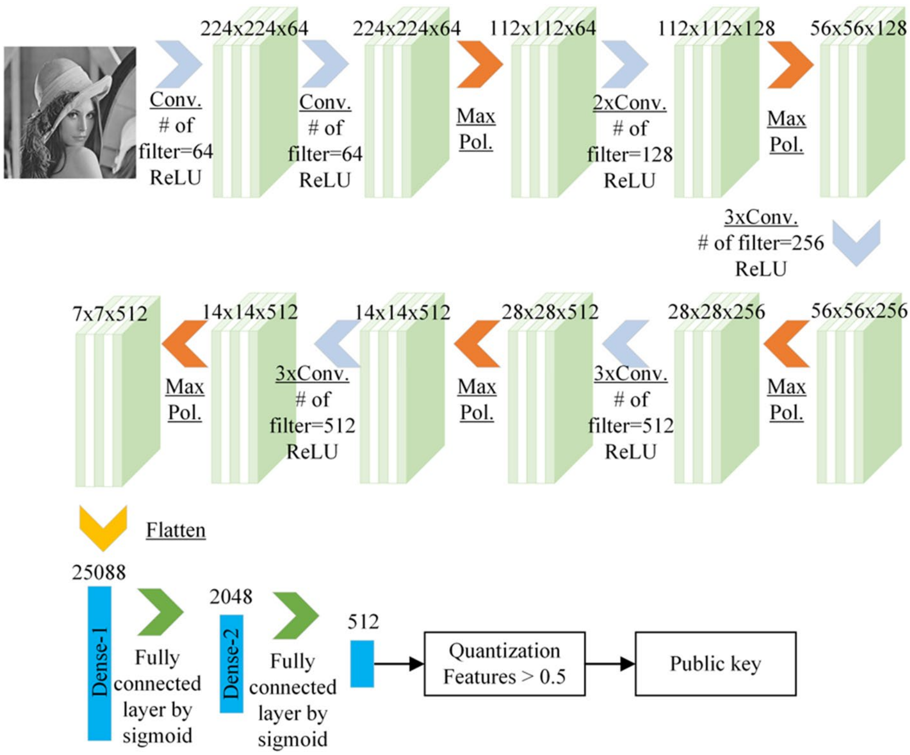

Fig. 2 Proposed deep CNN model and key generation procedure

a small-sized effective baseline architecture. The baseline model's depth, width and resolution are scaled by fixed coefficients. Thus, higher classification accuracy can be achieved with fewer parameters. EfficientNet also has some variants, i.e., it has approximately 8 times fewer parameters than the Resnet152 but about 6 times faster. BiT, which has its own variables, is actually a scalable variant of ResNet. The largest one is based on ResNet152. It was trained over JFT dataset having 300 M images [46].

For image encryption, the simplicity and speed of the architecture are more critical than the other applications. VGG16 architecture is not complex as compared to the others. Therefore, VGG16 seems to be sufficient for key generation. The architecture is implemented using Keras library's Tensorflow backend [9]. The CNN model consists of five layers with convolution and pooling processes. The dimension of the extracted features for each image is 7×7 with 512 channels. Hence, the number of features is $7 \times 7 \times 512 = 25{,}088$ that converted to a one-dimensional vector using a flatten layer added to the model. After flattening, there are two additional layers referred to as Dense-1 and Dense-2 functioning for dimension reduction. The dense layers' parameters are randomly initiated with Glorot-Uniform distribution that adjusts the network's initial weights [45].

Suppose that the number of inputs and outputs in a layer are denoted by n_j and n_{j+1}, respectively. In [45], it is proved that if the weights of a layer, denoted by $\boldsymbol{W}$, are initialized uniformly as

$$\boldsymbol{W} = U\left[\sqrt{\frac{6}{n_j + n_{j+1}}}, -\sqrt{\frac{6}{n_j + n_{j+1}}}\right] \tag{1}$$

the variance of the layers' outputs is close to the input variance. By random initialization, the model generates a different public key for the same image at each time. Sigmoid is used as the activation function for every dense layer. Let $\boldsymbol{F}$ denotes the outputs of the flatten layer, and those for Dense-1 and Dense-2 as $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$, respectively. Bias initializers are set to zero. Thus, the outputs of the Dense layers are obtained as follows. $\boldsymbol{D_1} = \boldsymbol{\delta}(\boldsymbol{W_1 F})$ and $\boldsymbol{D_2} = \boldsymbol{\delta}(\boldsymbol{W_2 D_1})$, where δ denotes the sigmoid activation, $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ denotes randomly initialized weights for Dense-1 and Dense-2, respectively. Image features herewith get a value between 0 and 1 before being converted to binary. Binarization is achieved by comparing the output of the Dense-2 with 0.5. That is a public key is obtained as $\boldsymbol{P} = \boldsymbol{D_2} > \mathbf{0.5}$. The dimension of the Dense-2 layer determines the size of the public key.

3.2 The proposed chaotic map: log-map

Chaotic maps based on various logistic equations are frequently employed in image encryption schemes. The chaotic maps are used to generate a diverse sequence in accordance with the control parameter and initial value. The pixels of the image to be encrypted are herewith scrambled and manipulated using the produced sequence. To explain the working principle of the chaotic maps, conventional 1D logistic map (Eq. 2) and its Lyapunov exponent (LE) (Eq. 3) can be given as follows:

$$v_{i+1} = uv_i\left(1 - v_i\right), v_i \in (0, 1) \tag{2}$$

$$LE = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n-1} \ln |u - 2uv_i| \tag{3}$$

Here, $u \in [0, 4]$ is the control parameter (growing rate), v_i is the initial value, and n is the number of iterations for a specific value of the control parameter.

Bifurcation diagram, a plot of output values versus the map's control parameter, shows a multi-diverse solution suddenly appears while the control parameter changes. These solutions are also called bifurcation points [39]. Moreover, the LE, first mentioned in [8], is exploited to evaluate a chaotic map's performance regarding the system's predictability and sensitivity to the control parameter and initial value. Besides, the LE should be positive; the higher LE is, the better the chaotic characteristic shows.

The bifurcation diagram of the logistic map provided in Eq. (2) and its LE plot given in Eq. (3) are illustrated in Fig. 3a and b, respectively. It is seen that the LE is positive only for $u \in [3.57, 4)$ and hence the logistic map does not behave chaotically except for this range. Because the chaotic range should be larger for a more secure image encryption scheme, we propose a novel chaotic map particularly denoted log-map in Eq. (4) with a convenient LE equation in Eq. (5).

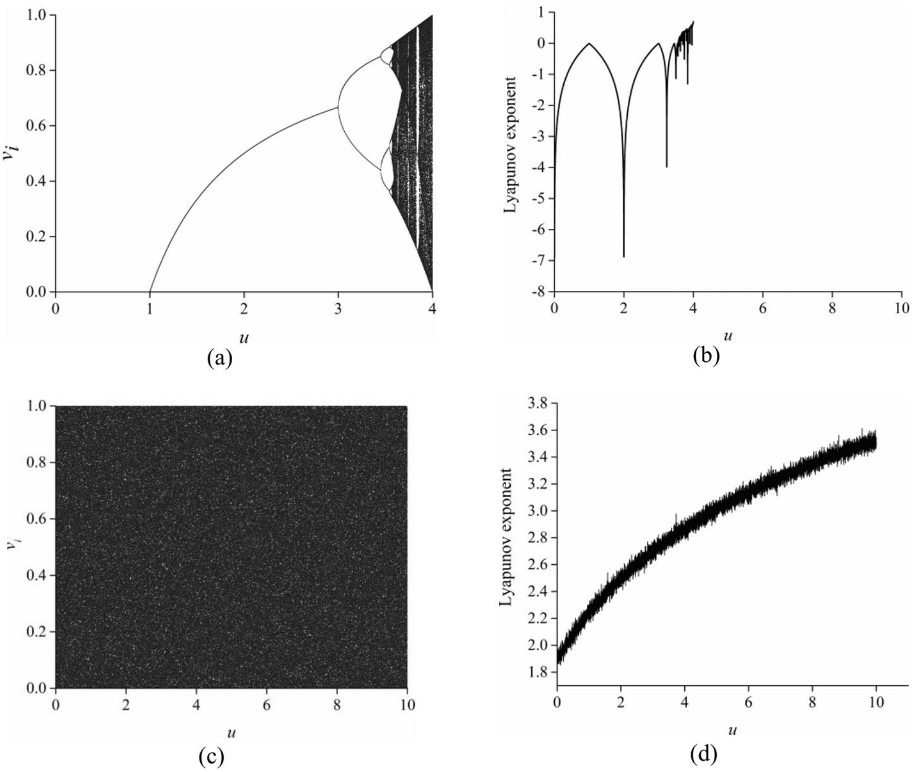

Fig. 3 Bifurcation sequence and LE plot (**a**) Bifurcation diagram of the logistic map, (**b**) LE of the logistic map, (**c**) Bifurcation diagram of the log-map, (**d**) LE of the log-map

$$v_{i+1} = \mathrm{mod}\left((u+e)\ln v_i, 1\right), v_i \in (0,1) \tag{4}$$

$$\mathrm{LE} = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n-1}\ln\left(\frac{u+e}{v_i}\right), (u+e)\ln v_i > \mathrm{floor}\left((u+e)\ln v_i\right) \tag{5}$$

Here, $u \in [0,\infty)$ is the control parameter, v_i is the initial value, n is the number of iterations for a specific value of the control parameter, and the floor is the greatest integer function.

The bifurcation diagram of the log-map and the relevant LE are plotted in Fig. 3c and d, respectively. The figures manifest that the LE value is positive for all nonnegative $u \in \mathbb{R}$. The log-map produces a chaotic map with higher ergodicity and diversity for a larger range of the control parameter besides a higher LE value.

3.3 DNA rules and encoding

DNA encoding is used in permutation and diffusion to effectively improve the security of image encryption. It increases especially key-space and the success of permutation and diffusion with the use of a dynamic chaotic map. Therefore, DNA provides resistance against differential attacks. It may be impossible to decrypt the image illegally. It would be harder to predict the encryption algorithm if another encoding technique is employed in addition to DNA. That is why DNA encoding is included in the log-map-based image encryption scheme to enhance entropy and correlation thanks to its ergodicity. DNA, which is indeed a biological term, is a molecule that carrying out genetic codes for surviving and continuing their generations. DNA consists of four nucleotides, adenine (A), thymine (T), cytosine (C), and guanine (G). In DNA rules, a binary number with two digits stands for each nucleotide, e.g., A-00, C-01, G-10, and T-11. Therefore, a code with a variety of 24 combinations can be obtained by using this rule. As shown in Table 1, these combinations can be reduced to eight type encoding rules according to Watson-Crick complementary rule [56]. Every grayscale pixel is illustrated with 8 bits. It corresponds to a DNA array with four nucleotides. For instance, the 167-pixel value is defined as a 10100111 binary array and it can be encoded as 00001110 by utilizing rule-4, which is also AATG in terms of DNA nucleotide. Thanks to this simple and effective encoding strategy, the DNA rule is successfully exploited to achieve a secure encryption scheme by manipulating the pixel values of the image.

Table 1 DNA encoding rules

Rule	0	1	2	3	4	5	6	7
00 (A)	A	A	C	G	C	G	T	T
01 (C)	C	G	A	A	T	T	C	G
10 (G)	G	C	T	T	A	A	G	C
11 (T)	T	T	G	C	G	C	A	A

4 Processes of the proposed encryption scheme

This section presents a novel image encryption scheme based on the log-map and provides some basic notions. The proposed encryption scheme is seriatim processed through four encrypting operations: permutation, DNA encoding, diffusion, and bit reversion. Throughout this paper, let $A := [a_{ij}]_{m\times n}$ be a gray-scale image matrix such that $i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$, a_{ij} is an unsigned integer number, $0 \le a_{ij} \le 255$, and $\text{floor}(A) := [\text{floor}(a_{ij})]$. Moreover, let $\hat{a}_{ij}$ denote a binary form of a_{ij}. Then, $\hat{A} := \left[\hat{a}_{ij}\right]_{m\times n}$ is referred to as the converted matrix of A to base 2.

Definition 2.1 Let $A := [a_{ij}]_{m\times n}$ and $B := [b_{ij}]_{m\times n}$ be two image matrices. Then, C is called the binary sum of A and B and is denoted by $C = A \oplus B$, if $C := \text{mod}\left(\hat{A} + \hat{B}, 2\right)$.

4.1 Obtaining the initial values and control parameters

In the encryption scheme, firstly, the public key is generated using the deep CNN following the procedure expressed in Section 3.1. Then, the initial value v and the control parameter u are obtained to produce chaotic sequences through the log-map that employs the public and secret keys. The chaotic sequences are utilized in the encrypting operations: permutation, DNA encoding, diffusion, and bit reversion. The production of the initial values and control parameters is shown in Algorithm 1 and even elaborately

Public Key C	0 1 1 1 1 1 1 1 … 0 0 0 1
Secret Key D	0 1 0 1 0 1 1 0 … 0 0 0 1
Main Key $E := \text{mod}(C + D, 2)$	0 0 1 0 1 0 0 1 … 0 0 0 0
Reshaped E has order 8×8×8	$\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 \end{bmatrix}$ … $\begin{bmatrix} 1 & 1 & \cdots & 0 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & 0 \end{bmatrix}$
$F := \text{mod}(\text{sum}(E, 2), 2)$ has order 8×1×8	$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 0 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$ … $\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$
Reshaped F has order 8×8	$\begin{bmatrix} 1 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & 0 \end{bmatrix}$
$S := [128\ 64\ 32\ 16\ 8\ 4\ 2\ 1]F$	[210 37 163 66 189 110 87 138]
Initial Values $V := [v_1\ \ v_2\ \ v_3\ \ v_4]$	[210 37 163 66]/256
Control Parameter $U := [u_1\ \ u_2\ \ u_3\ \ u_4]$s	[189 110 87 138]/256 + mod([189 110 87 138],10)

Fig. 4 Procedure to obtain the initial values and control parameters for processes of the encryption scheme

depicted in Fig. 4 through an illustrative example of the Lena image. The algorithm manages the generation of the public key referred to as binary matrix C via deep CNN in Steps 1–3. Once a secret key matrix D is constructed in Step 4, the main key matrix E is acquired by XOR operation between the public and secret keys in Step 5, and then the main key is divided into eight matrix groups each of which is used to produce the initial values and control parameters. In Step 6, the columns of each matrix group in self have imposed in turn a series of transaction mod(sum(E, 2), 2) then the results of each group 8×1 are combined to be F matrix size 8×8. In Step 7, the binary matrix F is afterward converted to a decimal matrix S. Finally, the initial values $V := [v_1 \ v_2 \ v_3 \ v_4]$ and control parameters $U := \left[u_1 \ u_2 \ u_3 \ u_4 \right]$ are respectively obtained by transactions $\frac{s_{1i}}{256}$ and $\frac{s_{1,i+4}}{256} + \text{mod}\left(s_{1,i+4}, 10\right)$ in Steps 8 and 9.

Algorithm 1: Obtaining initial values and control parameters

Step 1. Read an image matrix A

Step 2. Obtain a fuzzy row matrix $B \coloneqq \left[b_{1p}\right]_{1\times512}$by applying deep CNN to A

Step 3. Obtain the public key $C \coloneqq \left[c_{1p}\right]_{1\times512}$ defined by $c_{1p} \coloneqq \begin{cases} 1, & b_{1l} > 0.5 \\ 0, & \text{otherwise} \end{cases}$

Step 4. Construct a secret key $D \coloneqq \left[d_{1p}\right]_{1\times512}$ being a binary row matrix

Step 5. Evaluate the main key $E \coloneqq C$ XOR D and reshape E to 8×8×8 in shape

Step 6. Compute $F \coloneqq \text{mod}(\text{sum}(E,2),2)$ and then reshape F to 8×8 in shape where $\text{sum}(E,2)$ means the sum of the rows of each submatrix in shape 8×8 of E

Step 7. Figure out $S \coloneqq [128\ 64\ 32\ 16\ 8\ 4\ 2\ 1]F$

Step 8. Obtain the initial values $v_i \leftarrow \frac{s_{1i}}{256}$, $i \in \{1,2,3,4\}$

Step 9. Obtain the control parameters $u_i \leftarrow \frac{s_{1,i+4}}{256} + \text{mod}(s_{1,i+4}, 10)$, $i \in \{1,2,3,4\}$

4.2 The operations of the encryption scheme

The proposed scheme conducts the encryption of an image through Algorithm 2 consisting of four encrypting operations: permutation, DNA encoding, diffusion, and bit reversion. The processes of the encryption scheme are presented in Fig. 5 with an illustrative example for a 5×5 pixel sample of the Lena image. The pixels are scrambled and manipulated using the operations according to the chaotic sequence produced by the log-map. The obtained initial value and control parameter pairs are herein utilized for each operation to produce chaotic sequences in Steps 1, 4, 8, and 11, i.e., v_1 and u_1 pair is for the permutation, v_2 and u_2 pair is for the DNA encoding, v_3 and u_3 pair is for the diffusion, v_4 and u_4 pair is for the bit reversion. In the permutation (Steps 2 and 3), the position of every pixel is transferred with regard to the ascending order of the chaotic sequence. In this way, the position of every pixel is scrambled under the management of the main key and log-map. The pixel values are seriatim manipulated in the remaining three operations. In the DNA encoding (Steps 5–7), the chaotic sequence is exploited to determine the DNA rule from Table 1. The value of every pixel is changed in accordance with the determined DNA rule. In diffusion (Steps 9 and 10), the encoded pixels are undergone an XOR operation with a matrix Y_3 attained via the chaotic sequence. The last process, the bit reversion, is a new operation proposed in this study to improve the security further. In the bit reversion (Steps 12–14), diffused pixels are incurred an XOR operation with a matrix Z

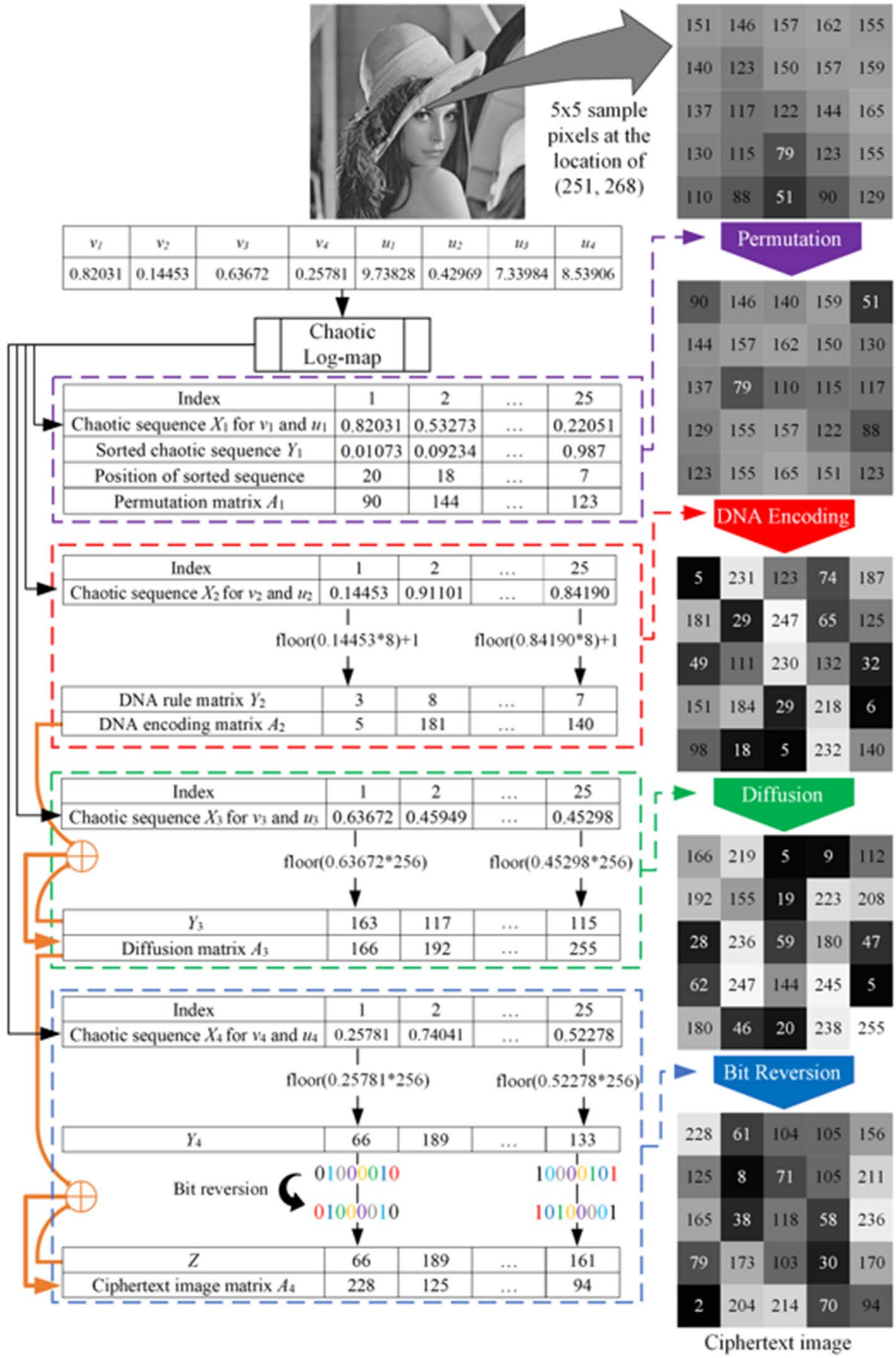

Fig. 5 Processes of the proposed encryption scheme with an illustrative example

acquired using the reverted bits of a matrix Y_4 obtained by the chaotic sequence. The outcome matrix A_4 of the bit reversion operation is thus the final ciphertext image.

Algorithm 2: The steps for the operations of the proposed image encryption scheme

Permutation

Step 1. Compute the first chaotic sequence $X_1 := [x_{1r}^1]_{1\times mn}$ by using the initial value v_1, the control parameter u_1, and the log-map (see Eq. (4))

Step 2. Obtain $Y_1 := [y_{1r}^1]_{1\times mn}$ by sorting X_1 in ascending order

Step 3. Compute a Permutation matrix $A_1 := [a_{kl}^1]_{m\times n}$ defined by $a_{kl}^1 := a_{ij}$ such that $x_{1,n(i-1)+j}^1 = y_{1,n(k-1)+l}^1$

DNA Encoding

Step 4. Compute the second chaotic sequence $X_2 := [x_{1r}^2]_{1\times mn}$ by using the initial value v_2, the control parameter u_2, and the log-map

Step 5. Evaluate the row matrix $Y_2 := [y_{1r}^2]_{1\times mn}$ defined by $Y_2 := 1 + \text{floor}(8X_2)$

Step 6. For all k and l, convert a_{kl}^1 to base-2 and rearrange these converted entries according to the rules that correspond to the values y_{1r} (see Table 1)

Step 7. For all k and l, convert the rearranged entries to base-10 and construct the DNA Encoding matrix $A_2 := [a_{kl}^2]_{m\times n}$

Diffusion

Step 8. Compute the third chaotic sequence $X_3 := [x_{1r}^3]_{1\times mn}$ by using the initial value v_3, the control parameter u_3, and the log-map

Step 9. Evaluate the row matrix $Y_3 := [y_{1r}^3]_{1\times mn}$ defined by $Y_3 := \text{floor}(256X_3)$ and then reshape Y_3 to $m\times n$ in shape

Step 10. Compute the Diffusion matrix $A_3 := [a_{kl}^3]_{m\times n}$ defined by $A_3 := A_2 \oplus Y_3$

Bit Reversion

Step 11. Compute the third chaotic sequence $X_4 := [x_{1r}^4]_{1\times mn}$ by using the initial value v_4, the control parameter u_4, and the log-map

Step 12. Evaluate the row matrix $Y_4 := [y_{1r}^4]_{1\times mn}$ defined by $Y_4 := \text{floor}(256X_4)$

Step 13. For all r, retype $\hat{y}_{1r}^4$ in reverse order and set to $\hat{z}_{1r}$. Then, reshape $Z := [z_{1r}]_{1\times mn}$ to $m\times n$ in shape

Step 14. Compute the Bit Reversion matrix $A_4 := [a_{kl}^4]_{m\times n}$ defined by $A_4 := A_3 \oplus Z$

5 Comparative performance analyses

The main aim of an encryption scheme is to improve its durability against potential cyber threats through network attacks such as denial-of-service, man-in-the-middle, and phishing. There are various cryptanalysis methods such as key-space, key sensitivity, entropy, histogram, correlation, differential attack, noisy attack, and cropping attack to evaluate the security of an encryption scheme by simulating some cyber threats [13, 53]. These cryptanalysis methods are performed on well-known images with a size of 512×512. In this study, the encryption and decryption processes are performed via MATLAB R2020b running on a workstation with I(R) Xeon(R) CPU E5–1620 v4 @ 3.5 GHz and 64 GB RAM. In order to generate hash codes, the VGG16-based CNN model is developed using Python 3.5. The scores of entropy, correlation, and differential attack are obtained on average for 200 run times. The outcomes of the proposed encryption scheme are even compared with some the-state-of-the-art results reported elsewhere [5, 6, 10, 12, 15, 22, 35, 54, 59, 60, 62, 63].

Table 2 A secret key and slightly one-bit changed versions

Key	Secret key
Key 1 (original)	01010110001110010111100100100100010000100010011001000101001010010100…1
Key 2 (changed)	01**1**0011000111001011110010010010001000010001001100100010100101001 0100…1
Key 3 (changed)	011**1**011000111001011110010010010001000010001001100100010100101001 0100…1
Key 4 (changed)	**1**0000110001110010111100100100100010000100010011001000101001010010100…1
Key 5 (changed)	100**1**0110001110010111100100100100010000100010011001000101001010010100…1

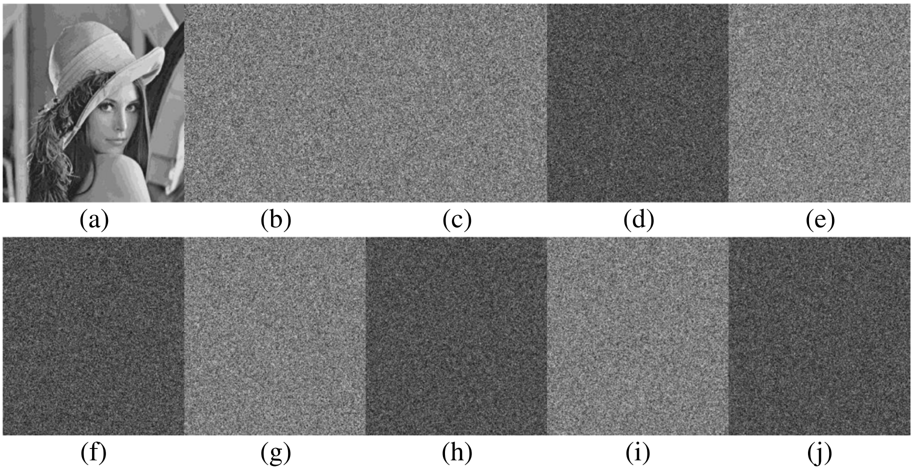

Fig. 6 Key sensitivity test: (**a**) plain image; (**b**) ciphertext with key 1; (**c**) ciphertext with key 2; (**d**) differential image between (b) and (c); (**e**) ciphertext with key 3; (**f**) differential image between (b) and (e); (**g**) ciphertext with key 4; (**h**) differential image between (b) and (g); (**i**) ciphertext with key 5; (**j**) differential image between (b) and (i)

5.1 Key-space analysis

Brute-force is a type of cyberattack based on predicting the key by trying numerous possible passwords or passphrases. An image encrypted with a short key is inherently vulnerable to this attack in time. If the key is longer, it will resist for a long time. Therefore, it would be impossible to guess the key if it has the proper length. The key-space analysis is utilized for testing the proof capability of the Brute-force attacks. According to this analysis, a key with longer than 2^{100} is considered for high security encryption [4]. In our scheme, we even propose an approach depending on deep CNN for generation SHA 512. Based on this key, eight floating numbers with 10^{15} precision is obtained to be used as the initial values and control parameters of the encrypting operations. Therefore, the keyspace of the proposed scheme is $10^{15\times 8} = 10^{120} \cong 2^{398}$. Notice that it is much higher than that of 2^{100}.

Table 3 Differences between cipher images produced by slightly different keys

Figures	One-bit changed key	Difference with 6(b) (%)
6(c)	Key 2	99.6250
6(e)	Key 3	99.6074
6(g)	Key 4	99.6265
6(i)	Key 5	99.6067
Mean		99.6164

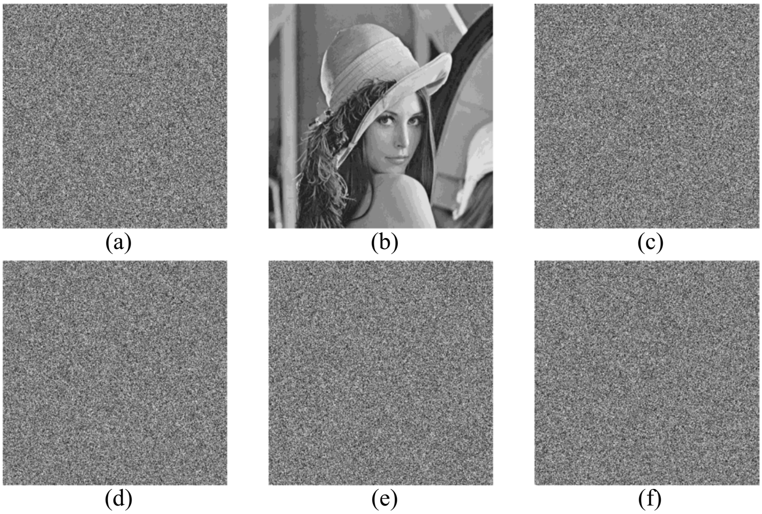

Fig. 7 Key sensitivity in the second case: (**a**) cipher image with *key 1*; (**b**) decipher image with *key 1*; (**c**) decipher image with *key 2*; (**d**) cipher image with *key 3*; (**e**) decipher image with *key* 4; (**f**) decipher image with *key* 5

5.2 Key sensitivity analysis

A secure encryption scheme should also be highly sensitive to the key, i.e., a slight change in the key must result in a significant variation in the image. Several tests can be performed in order to appreciate the key sensitivity. To this end, five secret keys are given in Table 2 contains an original key and its one-bit changed versions. The ciphertext images encrypted via those keys are illustrated in Fig. 6, and their differential images are presented to visually observe the number of pixels with the same tonal values. In the case of insensitive key, the pixels of differential images would be zero. Hence the same pixels with the same tonal values would seem black due to their zero difference values. As can be seen from Fig. 6d, (f), (h), and (j), there do not seem any black pixels in the differential images.

The distinctive level of the differential images with that of the original key in Fig. 6 is calculated to numerically evaluate the key sensitivity in Table 3. It is the non-zero-pixel percentage of differential images. The distinctive levels are respectively determined 99.6250%, 99.6074%, 99.6265% and 99.6067% for Key 2, 3, 4 and 5. Moreover, the mean of these levels is 99.6164%. It is evident from the results that the proposed encryption scheme is very sensitive to the key thanks to the CNN that is used to generate the public key.

Another test for measuring the key sensitivity is to decrypt the ciphertext image which is encrypted via the slightly changed keys. In this wise, it is aimed to visually examine the decrypted ciphertext images with one-bit changed keys so that whether they involve any information related to the plaintext image. Fig. 7 comparatively shows the decrypted images together with that of the ciphertext image with the original key in Fig. 7b. While the cyphertext image with the original key is exactly decrypted as the same plaintext image, the other images do not contain any texture from the plaintext image. This high sensitivity can be attributed to the CNN-based key production.

5.3 Histogram analysis

Histogram analysis presents a graphical distribution of the pixel value's number versus every tonal value in the image. Therefore, it allows investigating the uniformity of the image pixel's values. The manipulation performance of the encryption scheme can thus be evaluated through histogram analysis. The more uniform histogram, the better the manipulation performance. The histograms of the well-known plaintext images Lena, Cameraman, Baboon, Peppers, Airplane, and Barbara, and their ciphertext images are illustrated in Fig. 8. It can be clearly seen that the proposed image encryption scheme uniformly manipulates the tonal value of the images, implying that any information related to the plaintext image from the ciphertext images cannot be extracted. Therefore, the image encryption scheme can resist statistical attacks.

In order to further analyze the distribution of the pixels' tonal values, variance and χ^2 tests of the histogram are calculated. For a grayscale image, variance and χ^2 tests can be computed as follows:

$$var(\boldsymbol{X}) = \frac{1}{n^2}\sum_{i=0}^{n}\sum_{j=1}^{n}\frac{1}{2}\left(x_i - x_j\right)^2 \tag{6}$$

$$\chi^2 = \sum_{i=0}^{255}\frac{\left(n_i - n/256\right)^2}{n/256} \tag{7}$$

where n_i is the repetition frequency of a tonal value i and n is the number of total pixels. $n/256$ is hence the expected repetition frequency for each tonal value. $\boldsymbol{X}=\{x_1,x_2,\ldots,x_{256}\}$ is the vector of the histogram's tonal values. x_i and x_j are the numbers of pixels whose gray values are equal to i and j, respectively. For desiring high uniformity, the variance should be lowered as much as possible. On the other side, $\chi^2(0.05;255)$ should be lower than 293.25 for passing χ^2 test with 0.05, which is the significant level [64]. The results of variance and χ^2 test are tabulated in Table 4 for the images under the histogram analysis in Fig. 8. The proposed log-map-based image encryption scheme is hereby verified in terms of the results in Table 4.

5.4 Information entropy analysis

Information entropy is the most applied analysis to measure the uncertainty and disorderliness of a ciphertext image [65]. Therefore, it reflects the manipulation performance of an

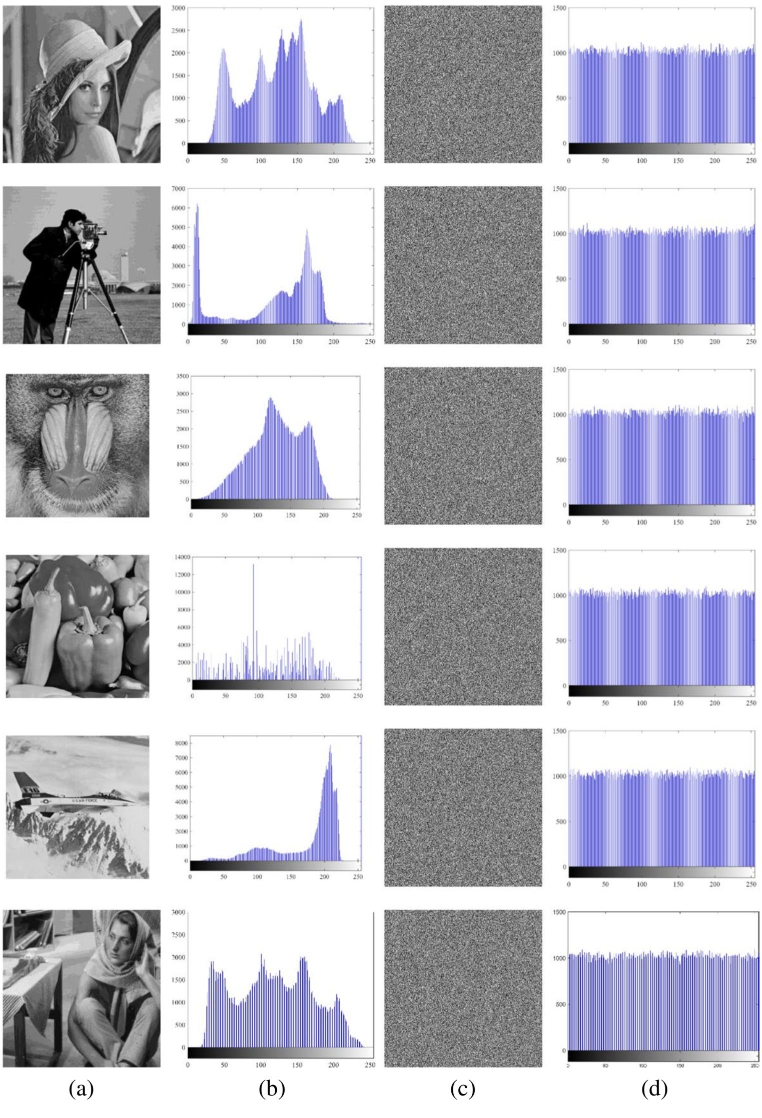

Fig. 8 Histograms of the well-known images: (**a**) the plaintext images, (**b**) histograms of the plaintext images, (**c**) the ciphertext images, (**d**) the histograms of ciphertext images

Table 4 Variance and χ^2 test results of the images under the histogram analysis

Test	Image	Lena	Cameraman	Peppers	Baboon	Barbara	Airplane
var	Plaintext	6,333,788.75	1,674,120.58	2,196,605.10	845,463.33	3,821,955.00	2,832,714.39
	Ciphertext	993.10	997.77	1007.18	988.73	980.32	1006.91
χ^2	Plaintext	158,344.71	418,530.14	549,151.27	211,365.83	95,548.87	708,178.59
	Ciphertext	247.30	248.46	250.81	252.81	245.60	250.74

Table 5 Information entropy of the images through the proposed log-map-based image encryption scheme

Image	Lena	Cameraman	Baboon	Peppers	Barbara	Airplane
Plaintext image	7.4455	7.0479	7.2925	6.7624	7.6321	6.7135
Ciphertext image	7.9994	7.9994	7.9994	7.9994	7.9993	7.9994

image encryption scheme. The information entropy of an image can be computed by the following equation:

$$H(x) = \sum_{i=0}^{2^n-1} p\left(x_i\right) \log_2 \frac{1}{p\left(x_i\right)} \tag{8}$$

Here, x is the information source. The probability of x_i can be represented by $p(x_i)$ and 2^n referring to the overall states.

The maximum entropy of an encrypted image can theoretically be 8 for a grayscale image. Therefore, the diffusion performance of an image encryption scheme is evaluated better than how the entropy is close to this theoretical value. The information entropy of the under-analysis images encrypted through the proposed scheme is given in Table 5. They are compared with the available state-of-the-art results [5, 6, 12, 15, 22, 59, 63] for Lena, Cameraman, Peppers, and Barbara ciphertext images in Table 6. As seen from Table 5, all information entropy values of the ciphertext images are closely near to 8. Moreover, the proposed scheme encrypts images with the best information entropy of 7.9994 among the other suggested schemes [5, 6, 12, 15, 22, 59, 63]. These cryptanalysis results mean that the proposed scheme provides the most assured images against cyberattacks.

5.5 Correlation analysis

Given that high correlation inherently exists among the adjacent pixels of a plaintext image. However, a secure image encryption scheme must alleviate the correlation by introducing an effective permutation operation. The correlation coefficient of an image can be computed in horizontal, vertical, and diagonal directions using the following equation:

$$r_{xy} = \frac{E[x - E(x)]\left[y - E(y)\right]}{\sqrt{D(x)\sqrt{D(y)}}} \tag{9}$$

Table 6 Comparative information entropy of several ciphertext images of several methods

Ciphertext image	[12]	[22]	[5]	[6]	[59]	[15]	[63]	Proposed scheme
Lena	7.9993	7.9993	7.9975	7.9982	**7.9994**	7.9970	**7.9994**	**7.9994**
Cameraman	–	–	–	–	7.9993	7.9973	7.9970	**7.9994**
Peppers	–	**7.9994**	–	–	7.9993	7.9969	–	**7.9994**
Barbara	7.9992	–	7.9985	7.9981	–	–	–	**7.9993**

Table 7 Correlation coefficients of the ciphertext images under-analysis

Direction	Lena	Cameraman	Baboon	Peppers	Barbara	Airplane
Horizontal	-29×10^{-5}	27×10^{-5}	52×10^{-7}	-80×10^{-6}	86×10^{-6}	47×10^{-6}
Vertical	21×10^{-5}	45×10^{-6}	36×10^{-5}	12×10^{-6}	18×10^{-6}	75×10^{-6}
Diagonal	33×10^{-6}	-65×10^{-6}	10×10^{-5}	18×10^{-6}	-19×10^{-6}	12×10^{-6}

where the two auxiliary equations are $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$ and $D(x) = \frac{1}{N}\sum_{i=1}^{N}\left[\left(x_i - E(x)\right]^2$. Here, x_i and y_i are tonal values of *i-th* pair of the selected adjacent pixels, and N represents the number of the pixel samples. For the correlation analysis performed in this study, $N=3000$ pixel samples are randomly selected from the ciphertext image.

The correlation coefficients of the under-analysis images encrypted through the proposed scheme are given in Table 7, and they are also compared with the available correlation coefficients in the literature for Lena, Cameraman, Peppers, and Barbara ciphertext images in Table 8 [5, 6, 12, 15, 22, 59, 63]. From Table 7, the proposed log-map-based image encryption scheme reduces the correlation coefficients to 96.955×10^{-6} which is almost zero. Besides, with 12×10^{-66} for Peppers, it outperforms the other schemes suggested in the literature in terms of the correlation coefficients tabulated in Table 8.

The correlation distribution of two pixels of Lena's plaintext and ciphertext images are illustrated in Fig. 9 for horizontal, vertical, and diagonal directions. Since the correlation distribution of a mono-color image would be a point and that of completely correlated pixels would be on $y=x$ line. While the correlation coefficients of Lena's plaintext image for the horizontal, vertical, and diagonal directions are 0.9737, 0.9838, and 0.9645, respectively; their correlation distributions are mainly concentrated on $y=x$ line. On the other hand, the correlation distributions of the ciphertext image are uniformly spread over the space because of their very low correlation coefficients of -29×10^{-5}, 21×10^{-5}, and 33×10^{-6} (see Table 7).

5.6 Differential attack analysis

A differential attack attempts to learn the key and figure out the encryption scheme by tracing the differences. The differential analysis examines the encryption scheme counter the cyberattacks through the difference between two ciphertext images one of which bits is changed in the plaintext image. It has capable of evaluating both permutation and manipulation performances of an encryption scheme. Therefore, a reliable encryption scheme should be sensitive to a slight change in the plaintext image [52]. Differential attack analysis is carried out by calculating the following number of pixels changing rate (NPCR) and the unified average changing intensity (UACI).

$$D(i,j) = \begin{cases} 0, if\ C^1(i,j) = C^2(i,j) \\ 1, if\ C^1(i,j) \neq C^2\ (i,j) \end{cases} \tag{10}$$

$$NPCR = 100\frac{\sum_{i,j} D(i,j)}{mn} \tag{11}$$

Table 8 Comparative correlation coefficients

Ciphertext image	Direction	[12]	[22]	[5]	[6]	[59]	[15]	[63]	Proposed scheme
Lena	Horizontal	13×10^{-3}	10×10^{-4}	27×10^{-4}	14×10^{-4}	32×10^{-4}	22×10^{-4}	73×10^{-5}	$\mathbf{-29\times10^{-5}}$
	Vertical	17×10^{-3}	-15×10^{-4}	13×10^{-4}	14×10^{-4}	16×10^{-4}	13×10^{-4}	44×10^{-5}	$\mathbf{21\times10^{-5}}$
	Diagonal	67×10^{-5}	26×10^{-4}	11×10^{-4}	12×10^{-4}	23×10^{-4}	8×10^{-4}	36×10^{-5}	$\mathbf{33\times10^{-6}}$
Cameraman	Horizontal	–	–	–	–	14×10^{-4}	39×10^{-4}	93×10^{-4}	$\mathbf{27\times10^{-5}}$
	Vertical	–	–	–	–	2×10^{-4}	7×10^{-4}	10×10^{-5}	$\mathbf{45\times10^{-6}}$
	Diagonal	–	–	–	–	35×10^{-4}	86×10^{-4}	31×10^{-4}	$\mathbf{-65\times10^{-6}}$
Peppers	Horizontal	–	7×10^{-5}	–	–	6×10^{-4}	1×10^{-4}	–	$\mathbf{-80\times10^{-6}}$
	Vertical	–	43×10^{-4}	–	–	38×10^{-4}	-26×10^{-4}	–	$\mathbf{12\times10^{-6}}$
	Diagonal	–	-18×10^{-4}	–	–	10×10^{-4}	-23×10^{-4}	–	$\mathbf{18\times10^{-6}}$
Barbara	Horizontal	-70×10^{-4}	–	-13×10^{-4}	12×10^{-4}	–	–	–	$\mathbf{86\times10^{-6}}$
	Vertical	-79×10^{-4}	–	43×10^{-4}	27×10^{-4}	–	–	–	$\mathbf{18\times10^{-6}}$
	Diagonal	-22×10^{-3}	–	10×10^{-4}	-10×10^{-4}	–	–	–	$\mathbf{-19\times10^{-6}}$

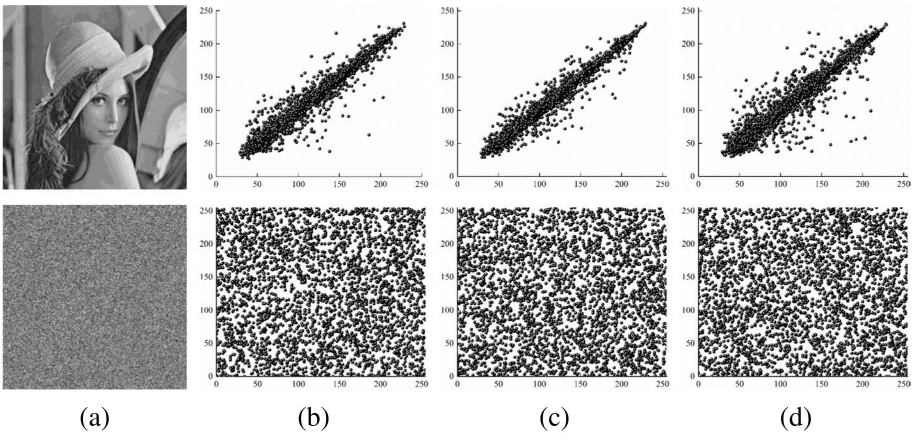

Fig. 9 (**a**) the image under-analysis; the correlation distribution of two adjacent pixels for three directions: (**b**) horizontal, (**c**) vertical, (**d**) diagonal

Table 9 Scores of NPCR and UACI for under-analysis images encrypted through the proposed scheme

Direction	Lena	Cameraman	Baboon	Peppers	Barbara	Airplane
NPCR	99.6094	99.6087	99.6094	99.6085	99.6085	99.6082
UACI	33.4622	33.4578	33.4652	33.4659	33.4631	33.4607

Table 10 Comparative scores of NPCR and UACI

Image	Test	[12]	[22]	[5]	[6]	[59]	[15]	[63]	The proposed scheme
Lena	NPCR	99.5800	99.6000	99.6912	99.6621	99.6000	99.60934	99.6078	99.6094
	UACI	33.4300	33.4700	33.5098	33.5278	33.5000	33.45969	33.4268	33.4622
Cameraman	NPCR	–	–	–	–	99.6000	99.60683	99.6323	99.6087
	UACI	–	–	–	–	33.5500	33.44610	33.4096	33.4578
Peppers	NPCR	–	0.9960	–	–	99.6100	99.60576	–	99.6085
	UACI	–	33.4600	–	–	33.5200	33.50204	–	33.4659
Barbara	NPCR	99.6100	–	99.6912	99.7501	–	–	–	99.6085
	UACI	33.4300	–	33.5098	33.5102	–	–	–	33.4631

$$UACI = 100\frac{1}{mn}\left[\sum_{i,j}\frac{\left|C^1(i,j) - C^2(i,j)\right|}{255}\right] \tag{12}$$

where m and n denote the height and width of the image. C^1 and C^2 are the ciphertext images before and after 1-bit of the plaintext image is altered, respectively. For a 1-bit altered grayscale image, the ideal scores of NPCR and UACI are expected to be 99.6094% and 33.4635%, respectively [58]. The scores of NPCR and UACI for under-analysis images

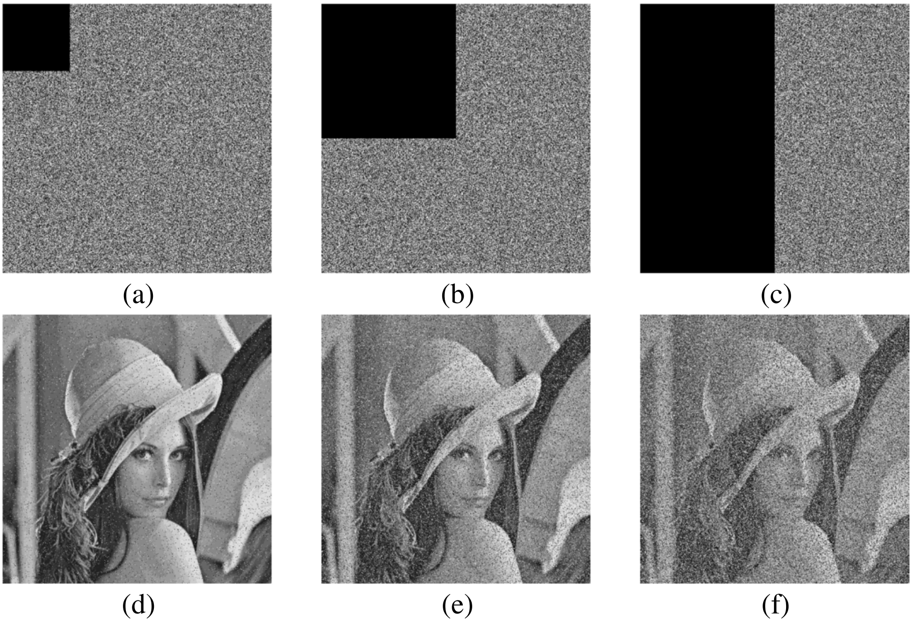

Fig. 10 Cropping attack analysis of the proposed encryption scheme over the Lena image, ciphertext images cropped with ratios (**a**) 1/16, (**b**) 1/4, (**c**) 1/2, and decrypted images with ratios (**d**) 1/16, (**e**) 1/4, (**f**) 1/2

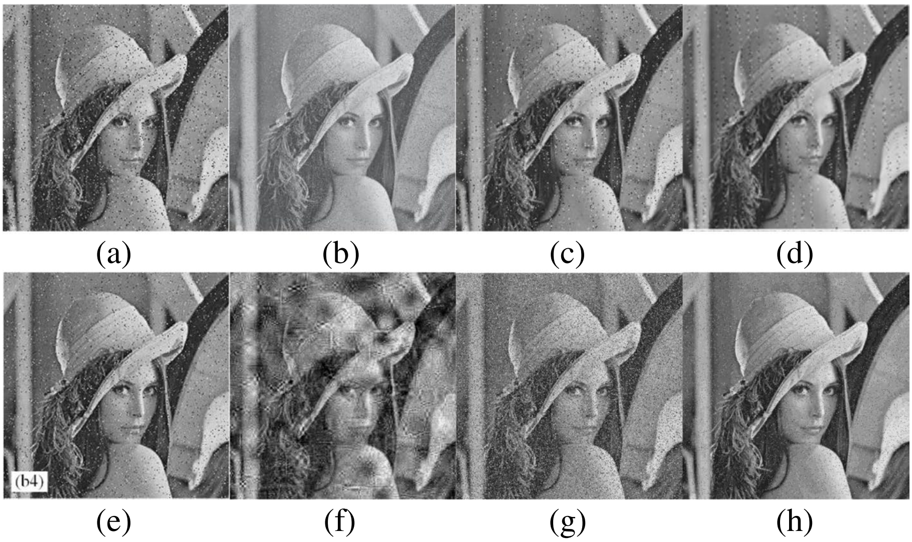

Fig. 11 Comparative cropping attack analysis of the decrypted Lena image cropped with 1/16 ratio (**a**) [15], (**b**) [63], (**c**) [62], (**d**) [60], (**e**) [35], (**f**) [10], (**g**) [54], (**h**) The proposed scheme

Table 11 Results of the PSNR metric for under-analysis images decrypted through the proposed scheme

Cropping ratio	Lena	Cameraman	Baboon	Pepper	Barbara	Airplane
1/16	21.37	20.55	21.61	20.92	20.87	20.14
1/4	15.25	14.44	15.64	14.89	14.85	14.08
1/2	12.23	11.43	12.65	11.84	11.82	11.08

Table 12 The comparative results of the PSNR metric

Image	Cropping ratio	[63]	[35]	The proposed scheme
Lena	1/16	17.58	20.78	21.37
	1/4	15.03	14.96	15.25
	1/2	12.13	12.08	12.23

encrypted through the proposed scheme are given in Table 9, and some of them are compared in Table 10 with the scores for Lena, Cameraman, Peppers, and Barbara that are available in the literature. It is confirmed that the proposed log-map-based scheme encrypts images with the closest results to the ideal scores.

5.7 Cropping attack analysis

Some parts of the ciphertext images can be lost or abused by the cyberattacks during the transmitting over the network. Cropping attack analysis assesses the competence of an encryption scheme regarding not only the permutation but also the manipulation performance. Hence, a robust and stable encryption scheme can recover the cropped image with the minimum degeneration [51]. In order to analyze the proposed encryption scheme, the encrypted Lena images cropped with ratios of 1/16, 1/4, and 1/2 are decrypted in Fig. 10, and the image cropped with 1/16 is also compared in Fig. 11 with the other schemes suggested in the literature. From the visual results, the proposed encryption scheme effectively recovers the cropped images with the least deterioration even if it is cropped with 1/2 as well as it is corroborated by the comparison.

Furthermore, the Lena image is numerically appreciated the proposed scheme by calculating Peak Signal to Noise Ratio (PSNR) defined in Eq. (13) precisely measures image quality based on comparing to the uncropped plaintext images [17]. Therefore, PSNR should be higher as much as possible for the lower degeneration. The results of PSNR for under-analysis images decrypted through the proposed scheme are listed in Table 11, and that for the Lena are compared with the existing results reported elsewhere in Table 12 [35, 63]. Thanks to the high PSNR, the cropping performance of the proposed encryption scheme is even validated in addition to the visual results in Fig. 11.

$$PSNR := 10 \log \left(\frac{255^2}{MSE} \right) \tag{13}$$

where MSE is the mean squared error and defined as

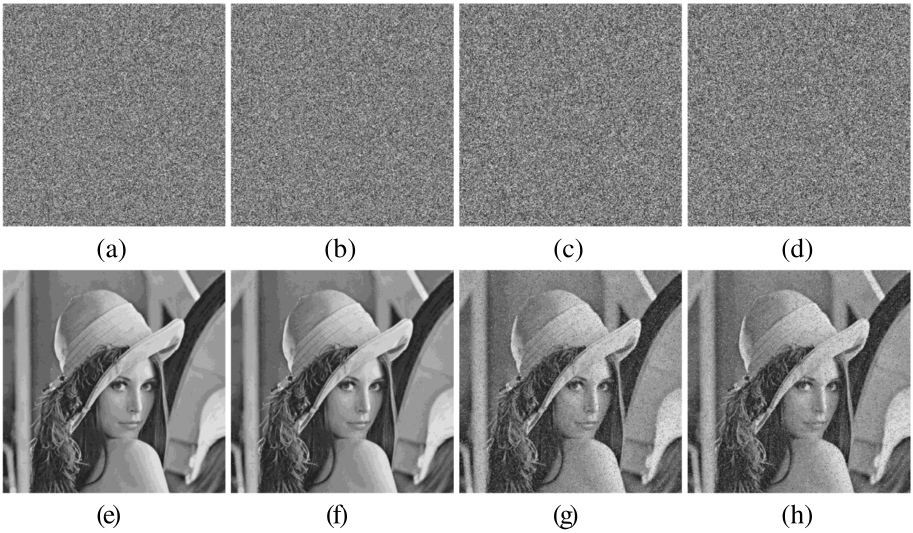

Fig. 12 Ciphertext images with adding various SPN densities of (**a**) 0.001, (**b**) 0.005, (**c**) 0.01, (**d**) 0.1 and their decrypted images with SPN densities of (**e**) 0.001, (**f**) 0.005, (**g**) 0.01, (**h**) 0.1

$$MSE := \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(e_{ij}-f_{ij}\right)^2 \tag{14}$$

where $E := [e_{ij}]$ is the plaintext image and $F := [f_{ij}]$ is the decrypted image that is cropped.

5.8 Noise attack analysis

Noise can be inherently added or inserted by the cyberattacks during the network transferring. Noise attack analysis is us ed to evaluate the permutation and manipulation performances of an image encryption scheme as similar to cropping attack analysis. Salt & pepper noise (SPN) is mostly utilized to inspect a scheme against noise attacks. In this way, restoring the ability of the scheme is investigated by adding SPN to the ciphertext image. Figure 12 visually demonstrates the decrypted versions of the ciphertext images by adding various SPN densities of 0.001, 0.005, 0.01, and 0.1. Moreover, these results are numerically verified via PSNR metric measured as 40.23, 32.65, 29.23, and 19.23 for images with SPN densities of 0.001, 0.005, 0.01, and 0.1, respectively. Consequently, the proposed log-map-based image encryption scheme maximally restores all images even if the image has a high SPN density such as 0.1.

5.9 Encryption processing time analysis

In addition to the cryptanalysis performed above, the processing time of an encryption scheme is an essential aspect of a realistic image encryption scheme. Our scheme processes 0.015 (s) and 0.3996 (s) for deep CNN and encrypting operations, respectively. In other words, it encrypts throughout an image in 0.4146 (s) and decrypts in 0.3662 (s). Therefore, it can be applied to real-time applications due to the fast processing time.

6 Conclusion

This study proposes an image encryption scheme, which depends on a novel chaotic log-map, key generation via deep CNN, and a new bit reversion for encrypting operations. The images are securely encrypted across four operations: permutation, DNA encoding, diffusion, and bit reversion in which the pixels are scrambled and manipulated. The diverse chaotic sequences for the operations are produced by the log-map whose initial values and control parameters are obtained using the produced public key and secret key. The performance of the encryption scheme is visually and numerically investigated for the well-known images concerning a variety of trusted cryptanalysis as well as validated by comparing with the available results in the literature. The superior cryptanalyses such as key-space, mean entropy, mean correlation, NPCR, UACI, encryption processing time are 2^{398}, 7.9994, 96.955×10^{-6}, 99.6087, 33.4624 and 0.4146 (s), respectively. Hence it is demonstrated that the proposed log-map-based image encryption scheme is prominent among the suggested schemes in the literature thanks to the superior results. The main limitation of the log-map is to be 1D. Although a 1D chaotic map is successful in bifurcation and LE tests, it might be possible to predict the trajectory of a 1D map. In the future work, it is planned to derive a multi-dimensional log-map and combinational maps with other effective chaotic maps like sine, Henon, Chebyshev, etc.

Acknowledgements This research was funded by University of Economics Ho Chi Minh City, Vietnam. Fund receiver: Dr. Dang Ngoc Hoang Thanh.

References

1. Abdulla AA, Sellahewa H, Jassim SA (2019) improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images. Multimed Tools Appl 7813(78):17799–17823. https://doi.org/10.1007/S11042-019-7166-7
2. Abdulla AA, Sellahewa H, Jassim SA (2014) Stego quality enhancement by message size reduction and Fibonacci bit-plane mapping. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 8893:151–166. https://doi.org/10.1007/978-3-319-14054-4_10
3. Alawida M, Samsudin A, Sen TJ et al (2019) A new hybrid digital chaotic system with applications in image encryption. Signal Process 160:45–58. https://doi.org/10.1016/j.sigpro.2019.02.016
4. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 16:2129–2151. https://doi.org/10.1142/S0218127406015970
5. Asgari-Chenaghlu M, Balafar MA, Feizi-Derakhshi MR (2019) A novel image encryption algorithm based on polynomial combination of chaotic maps and dynamic function generation. Signal Process 157:1–13. https://doi.org/10.1016/j.sigpro.2018.11.010
6. Asgari-Chenaghlu M, Feizi-Derakhshi MR, Nikzad-Khasmakhi N et al (2021) Cy: chaotic yolo for user intended image encryption and sharing in social media. Inf Sci (Ny) 542:212–227. https://doi.org/10.1016/j.ins.2020.07.007
7. Bao L, Yi S, Zhou Y (2017) Combination of Sharing Matrix and Image Encryption for Lossless (k,n)-Secret Image Sharing. IEEE Trans Image Process 26:5618–5631. https://doi.org/10.1109/TIP.2017.2738561
8. Briggs K (1990) An improved method for estimating Liapunov exponents of chaotic time series. Phys Lett A 151:27–32. https://doi.org/10.1016/0375-9601(90)90841-B

9. Capelo L (2018) Beginning application development with TensorFlow and Keras: learn to design, develop, train, and deploy TensorFlow and Keras models as real-world applications. Packt Publishing
10. Chai X, Zheng X, Gan Z et al (2018) An image encryption algorithm based on chaotic system and compressive sensing. Signal Process 148:124–144. https://doi.org/10.1016/j.sigpro.2018.02.007
11. Chai X, Fu X, Gan Z et al (2019) A color image cryptosystem based on dynamic DNA encryption and chaos. Signal Process 155:44–62. https://doi.org/10.1016/j.sigpro.2018.09.029
12. Chai X, Gan Z, Yuan K et al (2019) A novel image encryption scheme based on DNA sequence operations and chaotic systems. Neural Comput Appl 31:219–237. https://doi.org/10.1007/s00521-017-2993-9
13. Chai X, Bi J, Gan Z et al (2020) Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. Signal Process 176:107684. https://doi.org/10.1016/j.sigpro.2020.107684
14. Chen J, Chen L, Zhou Y (2020) Cryptanalysis of a DNA-based image encryption scheme. Inf Sci (Ny) 520:130–141. https://doi.org/10.1016/j.ins.2020.02.024
15. Chen C, Sun K, He S (2020) An improved image encryption algorithm with finite computing precision. Signal Process 168:1–10. https://doi.org/10.1016/j.sigpro.2019.107340
16. Chen X, Li L, Sharma A et al (2021) The Application of Convolutional Neural Network Model in Diagnosis and Nursing of MR Imaging in Alzheimer's Disease. Interdiscip Sci Comput Life Sci. https://doi.org/10.1007/S12539-021-00450-7
17. Enginoğlu S, Erkan U, Memiş S (2019) Pixel similarity-based adaptive Riesz mean filter for salt-and-pepper noise removal. Multimed Tools Appl 78:35401–35418. https://doi.org/10.1007/s11042-019-08110-1
18. Enayatifar R, Guimarães FG, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 115:131–140. https://doi.org/10.1016/j.optlaseng.2018.11.017
19. Erkan U, Toktas A, Toktas F et al (2022) 2D eπ-map for image encryption. Inf Sci (Ny). https://doi.org/10.1016/j.ins.2021.12.126
20. Farah MAB, Guesmi R, Kachouri A et al (2020) A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. Opt Laser Technol 121:105777. https://doi.org/10.1016/j.optlastec.2019.105777
21. Gao Y, Chen X, Ying Z, Parini C (2007) Design and performance investigation of a dual-element PIFA array at 2.5 GHz for MIMO terminal. IEEE Trans Antennas Propag 55:3433–3441. https://doi.org/10.1109/TAP.2007.910353
22. Hanis S, Amutha R (2019) A fast double-keyed authenticated image encryption scheme using an improved chaotic map and a butterfly-like structure. Nonlinear Dyn 95:421–432. https://doi.org/10.1007/s11071-018-4573-7
23. Haque AKMB, Bhushan B, Dhiman G (2021) Conceptualizing smart city applications: requirements, architecture, security issues, and emerging trends. Expert Syst https://doi.org/10.1111/EXSY.12753
24. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90
25. Houssein EH, Hussain K, Abualigah L et al (2021) An improved opposition-based marine predators algorithm for global optimization and multilevel thresholding image segmentation. Knowledge-Based Syst 229:107348. https://doi.org/10.1016/J.KNOSYS.2021.107348
26. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci (Ny) 480:403–419. https://doi.org/10.1016/j.ins.2018.12.048
27. Hua Z, Zhu Z, Yi S et al (2021) Cross-plane colour image encryption using a two-dimensional logistic tent modular map. Inf Sci (Ny) 546:1063–1083. https://doi.org/10.1016/j.ins.2020.09.032
28. Huang G, Liu Z, Van Der Maaten L et al (2017) Densely connected convolutional networks. Proc - 30th IEEE Conf Comput Vis Pattern Recognition, CVPR 2017 2017-January:2261–2269. https://doi.org/10.1109/CVPR.2017.243
29. Huo D, Zhu Z, Wei L et al (2021) A visually secure image encryption scheme based on 2D compressive sensing and integer wavelet transform embedding. Opt Commun 492:126976. https://doi.org/10.1016/J.OPTCOM.2021.126976
30. Kolesnikov A, Beyer L, Zhai X et al (2019) Big Transfer (BiT): General Visual Representation Learning. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 12350(LNCS):491–507
31. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems. pp. 1097–1105

32. Kumar V, Malik N, Dhiman G et al (2021) Scalable and storage efficient dynamic key management scheme for wireless sensor network. Wirel Commun Mob Comput 2021:. https://doi.org/10.1155/2021/5512879
33. Lan R, He J, Wang S et al (2018) Integrated chaotic systems for image encryption. Signal Process 147:133–145. https://doi.org/10.1016/j.sigpro.2018.01.026
34. Liu Y, Qin Z, Liao X, Wu J (2020) Cryptanalysis and enhancement of an image encryption scheme based on a 1-D coupled sine map. Nonlinear Dyn 100:2917–2931. https://doi.org/10.1007/s11071-020-05654-y
35. Luo Y, Lin J, Liu J et al (2019) A robust image encryption algorithm based on Chua's circuit and compressive sensing. Signal Process 161:227–247. https://doi.org/10.1016/j.sigpro.2019.03.022
36. Mansouri A, Wang X (2020) A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme. Inf Sci (Ny) 520:46–62. https://doi.org/10.1016/j.ins.2020.02.008
37. Namasudra S, Roy P, Vijayakumar P et al (2017) Time efficient secure DNA based access control model for cloud computing environment. Futur Gener Comput Syst 73:90–105. https://doi.org/10.1016/j.future.2017.01.017
38. Natarajan Y, Srihari K, Dhiman G et al (2021) An IoT and machine learning-based routing protocol for reconfigurable engineering application. IET Communications. https://doi.org/10.1049/CMU2.12266
39. Nusse HE, Yorke JA, Kostelich EJ (1994) Bifurcation diagrams. In: Dynamics: numerical explorations. Springer, New York, NY, pp 229–268
40. Oliva D, Esquivel-Torres S, Hinojosa S et al (2021) Opposition-based moth swarm algorithm. Expert Syst Appl 184:115481. https://doi.org/10.1016/J.ESWA.2021.115481
41. Russakovsky O, Deng J, Su H et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115:211–252. https://doi.org/10.1007/s11263-015-0816-y
42. Sahoo AK, Panigrahi TK, Dhiman G et al (2021) Enhanced emperor penguin optimization algorithm for dynamic economic dispatch with renewable energy sources and microgrid. J Intell Fuzzy Syst 40:9041–9058. https://doi.org/10.3233/JIFS-201483
43. Sewak M, Karim R, Pujari P (2018) Practical convolutional neural networks. Packt Publishing
44. Singh PD, Kaur R, Singh KD et al (2021) A novel ensemble-based classifier for detecting the COVID-19 disease for infected patients. Inf Syst Front 2021:1–17. https://doi.org/10.1007/S10796-021-10132-W
45. Sujatha E, Chilambuchelvan A (2018) Multimodal biometric authentication algorithm using Iris, palm print, face and signature with encoded DWT. Wirel Pers Commun 99:23–34. https://doi.org/10.1007/S11277-017-5034-1
46. Sun C, Shrivastava A, Singh S et al (2017) Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. 2017 IEEE International Conference on Computer Vision (ICCV) pp. 843–852. https://doi.org/10.1109/ICCV.2017.97
47. Suri S, Vijay R (2020) A Pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA. Neural Comput Appl 32:11859–11873. https://doi.org/10.1007/s00521-019-04668-x
48. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1-9. https://doi.org/10.1109/CVPR.2015.7298594
49. Talhaoui MZ, Wang X (2021) A new fractional one dimensional chaotic map and its application in high-speed image encryption. Inf Sci (Ny) 550:13–26. https://doi.org/10.1016/j.ins.2020.10.048
50. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning pp. 6105–6114.
51. Toktas A, Erkan U (2021) 2D fully chaotic map for image encryption constructed through a quadruple-objective optimization via artificial bee colony algorithm. Neural Comput & Applic. https://doi.org/10.1007/s00521-021-06552-z
52. Toktas A, Erkan U, Toktas F et al (2021) Chaotic Map Optimization for Image Encryption Using Triple Objective Differential Evolution Algorithm. IEEE Access 9:127814-127832. https://doi.org/10.1109/ACCESS.2021.3111691
53. Toktas A, Erkan U, Ustun D (2021) An image encryption scheme based on an optimal chaotic map derived by multi-objective optimization using ABC algorithm. Nonlinear Dyn 105:1885–1909. https://doi.org/10.1007/s11071-021-06675-x
54. Wang X, Gao S (2020) Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Inf Sci (Ny) 539:195–214. https://doi.org/10.1016/j.ins.2020.06.030
55. Wang H, Xiao D, Chen X et al (2018) Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map. Signal Process 144:444–452. https://doi.org/10.1016/j.sigpro.2017.11.005

56. Watson JD, Crick FHC (1953) Molecular structure of nucleic acids: a structure for Deoxyribose nucleic acid. Nature 171(4356):737–738 https://doi.org/10.1038/171737a0
57. Wen W, Wei K, Zhang Y et al (2020) Colour light field image encryption based on DNA sequences and chaotic systems. Nonlinear Dyn 99:1587–1600. https://doi.org/10.1007/s11071-019-05378-8
58. Wu Y, Noonan JP, Agaian S (2011) NPCR and UACI randomness tests for image encryption. Cyber J: Multi J Sci Technol J Selected Areas in Telecomm 4: 31–38
59. Wu J, Liao X, Yang B (2018) Image encryption using 2D Hénon-sine map and DNA approach. Signal Process 153:11–23. https://doi.org/10.1016/j.sigpro.2018.06.008
60. Wu Y, Zhang L, Qian T et al (2021) Content-adaptive image encryption with partial unwinding decomposition. Signal Process 181:107911. https://doi.org/10.1016/j.sigpro.2020.107911
61. Xuejing K, Zihui G (2020) A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. Signal Process Image Commun 80:1–11. https://doi.org/10.1016/j.image.2019.115670
62. Yang F, Mou J, Liu J et al (2020) Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application. Signal Process 169:1–16. https://doi.org/10.1016/j.sigpro.2019.107373
63. Yang Y, Wang L, Duan S et al (2021) Dynamical analysis and image encryption application of a novel memristive hyperchaotic system. Opt Laser Technol 133:106553. https://doi.org/10.1016/j.optlastec.2020.106553
64. Zhang X, Zhao Z, Wang J (2014) Chaotic image encryption based on circular substitution box and key stream buffer. Signal Process Image Commun 29:902–913. https://doi.org/10.1016/j.image.2014.06.012
65. Zhang F, Kodituwakku HADE, Hines JW et al (2019) Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. IEEE Trans Ind Informatics 15:4362–4369. https://doi.org/10.1109/TII.2019.2891261
66. Zheng P, Huang J (2018) Efficient encrypted images filtering and transform coding with Walsh-Hadamard transform and parallelization. IEEE Trans Image Process 27:2541–2556. https://doi.org/10.1109/TIP.2018.2802199
67. Zhou Y, Sharma A, Masud M et al (2021) Urban rain flood ecosystem design planning and feasibility study for the enrichment of smart cities. Sustain 13(9):5205. https://doi.org/10.3390/SU13095205