A Secure Image Encryption Scheme based on Fully-Connected-Like Neural Network and Edge Pixel Reset

Yaohui Sheng
Changchun University of Science and Technology
Jinqing Li (lijinqing@cust.edu.cn)
Changchun University of Science and Technology https://orcid.org/0000-0002-5580-2794
Xiaoqiang Di
Changchun University of Science and Technology
Zhenlong Man
Changchun University of Science and Technology
Zefei Liu
Changchun University of Science and Technology

Research Article

Keywords: Fully-Connected-Like Neural Network , Cyclic Shift Transformation , bit-level diffusion , bidirectional diffusion , Chaotic systems

Posted Date: April 6th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-345101/v1

Noname manuscript No.
(will be inserted by the editor)

A secure image encryption scheme based on Fully-Connected-Like Neural Network and edge pixel reset

Yaohui Sheng · Jinqing Li · Xiaoqiang Di · Zhenlong Man · Zefei Liu

Received: date / Accepted: date

Abstract When digital images are transmitted and stored in the currently open network environment, they often face various risks. A secure image encryption based on Fully-Connected-Like Neural Network (FCLNN) and edge pixel reset is proposed. Firstly, using random noise to reset the image last-bit of the edge pixels to generate different keys for each encryption. Secondly, the image rows and columns are transformed by Cyclic Shift Transformation (CST). Then, the image is diffused at the bit-level by using FCLNN. Finally, forward and reverse diffusions are performed on the image to generate the cipher image. In addition, the result of convolution operation between plain image and chaotic sequence is introduced to set the initial value of the chaotic system to establish the correlation between plain image and algorithm, which makes the algorithm resistant to known/chosen plaintext attack. The simulation results show that the proposed algorithm has negligible loss, and the decrypted image is visually identical to the original image. At the same time, the algorithm has a large key space, can resist common attacks such as statistical attacks, differential attacks, noise attacks, and data loss attacks, and has high security.

Keywords Fully-Connected-Like Neural Network · Cyclic Shift Transformation · bit-level diffusion · bidirectional diffusion · Chaotic systems

Y. Sheng · J. Li (✉) · X. Di · Z. Man · Z. Liu
School of Computer Science and Technology, Changchun University of Science and Technology, Jilin Province Key Laboratory of Network and Information Security, Information Center, Changchun University of Science and Technology, Changchun, 130033, China
E-mail: lijinqing@cust.edu.cn

1 Introduction

With the development of Internet technology, image information has an explosive growth. Images contain a large number of information with the intuitive visual effects and gradually become an important information carrier in social communication [1–3]. At the same time, image also plays an important role in medicine, military, aerospace, and other important fields [4–6]. However, with the impact of science and technology on people's lives, people also begin to worry about the security of image information in the transmission process. When the image is uploaded on the public channel, it may be maliciously damaged or illegally copied [7], so personal privacy, military secrets, and even national strategy are facing risks.

Therefore, effective methods are needed to protect image information. In the process of transmission, the direct method to protect the image is image encryption [8, 9], which encrypts the image into meaningless noise-like information. After receiving the image information, only a legal key can be used to obtain the correct decrypted image. Traditional encryption methods mostly convert images into bit streams and then encrypt them, such as DES [10], AES [11], IDEA [12], and so on. However, unlike bit streams, images have the characteristics of high correlation between adjacent pixels, large data capacity, and high redundancy [3]. Thus, traditional encryption methods may ignore these characteristics of the image, resulting in insufficient security of the image encryption algorithm [13–15].

Chaotic systems have the characteristics of initial value sensitivity, ergodicity and unpredictability [16], so it is very suitable for image encryption systems. Since chaos theory was first applied to image encryption [17], chaotic cryptosystem has attracted a large number of

scholars. At present, many image encryption algorithms based on chaos have appeared. Feifei Yang et al. proposed a new 4D fractional order laser chaotic system based on Lorenz Haken Model and applied it to image encryption. The chaotic sequence generated by the chaotic system has good pseudo randomness, and the encryption result also has good security [18]. Mingxu Wang et al. proposed an improved cross coupled spatiotemporal chaotic system and designed a bit-level image encryption algorithm based on the chaotic system, which has high security against common attacks [1].

However, some existing chaotic-based encryption algorithms have some shortcomings. Such as, the key design is simple [19, 20], cannot resist some plaintext attacks [21–23] and requires a large number of pseudo-random numbers [9, 23, 26]. These methods may lead to inefficient encryption processes and lower security levels. For instance, Wei Feng et al. proposed an image encryption method based on discrete logarithm and memristor system, in which the hash code of an ordinary image is directly used as the key, which may be conducive to cryptanalysis [24]. After that, Guodong Ye et al. proposed a pixel image scrambling encryption algorithm, but through cryptanalysis, it has been proved that this method cannot resist the chosen plaintext attack [25]. Although the chaotic system used in these methods can produce excellent pseudo-random sequences, the simple encryption algorithm based on the chaotic system is still insufficient to meet the higher requirements of image information security. In order to improve the ability of the algorithm to resist various attacks, we design an image edge pixel reset algorithm to generate plaintext-related keys. In the encryption process, random noise is used to reset the least-significant bits of the edge pixels of the image.When the same image is encrypted many times, a completely different cipher image can be generated each time. Experimental analysis shows that the method can resist all kinds of attacks.

With the development of cross-application of chaotic systems and multiple technologies, scholars have designed a variety of new image encryption algorithms. Guodong Ye et al. proposed an image encryption method based on compressed sensing and random number insertion, which uses compressed sensing to compress and encrypt the scrambled image, ensuring encryption security and improving the encryption efficiency [8]. Xingbin Liu et al. proposed a quantum image encryption method, which uses a novel quantum representation mode to represent the image, and then encrypts the image. This method has good computational complexity [27]. Jian Zhou et al. proposed a medical image ROI encryption method based on game theory, which realized the best balance between efficiency and security by using game theory [4]. From these image encryption algorithms, it can be seen that the combination of an excellent chaotic system and a good algorithm design can improve the encryption efficiency and the security of the encryption algorithm. In this paper, we design a bit-level image encryption method combined with FCLNN. FCLNN simulates the structure of a fully connected neural network and uses chaotic systems and random matrices to update the weight of the network. This algorithm can change the value of image pixels while scrambling the 8-bit binary number of image pixels. Since FCLNN is a reversible network, it can be used in symmetric encryption algorithms. Simulation results show that FCLNN is very suitable for image encryption.

After the above discussion, combined with the good security and reasonable computing cost of chaotic systems, this paper proposes a secure image encryption scheme based on FCLNN and edge pixel reset. Our algorithm consists of four parts. First, the edge pixel reset algorithm is used to generate the key. Second, the row and column cyclic shift transform is used to scramble the image. Then, the bit-level diffusion of the image is performed by using the FCLNN. Finally, the bidirectional diffusion of the image is performed. The cipher image is obtained by encrypting the image for more than n rounds. The main contributions of this study are as follows:

(1) A novel bit-level image diffusion method based on FCLNN is designed, the weights of the network are generated by a random matrix and chaotic sequences, and the improved S-box is used as the activation function.
(2) A method of generating plaintext-related keys based on the image edge pixel reset algorithm is proposed. Even if the same image is encrypted twice with the same user key, the encryption keys used in the encryption process are completely different, and the cipher images obtained are also completely different.
(3) A new image encryption algorithm based on CST scrambling, FCLNN bit-level diffusion and bidirectional diffusion is given. The algorithm has good encryption performance and can effectively resist various attacks.

The rest of this paper is arranged as follows. Section 2 introduces FCLNN. Section 3 describes the proposed image encryption algorithm in detail. In section 4, a lot of simulation experiments are carried out. Finally, the whole paper is summarized in section 5.

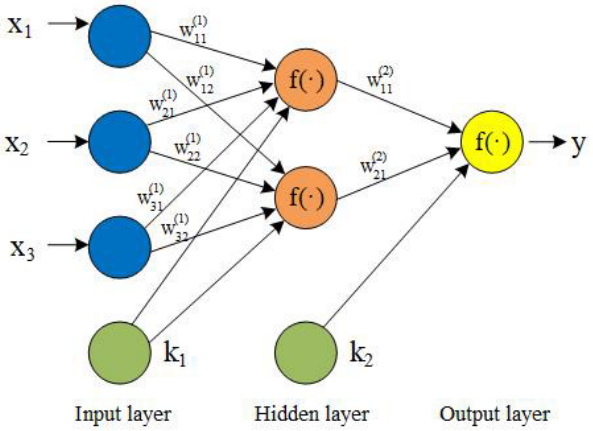

Fig. 1 The three-layer fully connected neural network

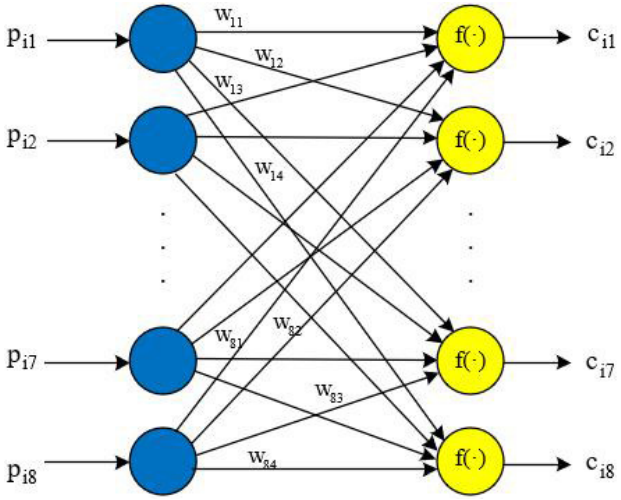

Fig. 2 The FCLNN

2 FCLNN

Neural network is a mathematical model or computational model that imitates the structure and function of biological neural network [28]. Fig. 1 shows a three-layer fully connected neural network with three inputs and one output, x_1, x_2, x_3 are the inputs of the fully connected neural network, y is the output of the fully connected neural network,$w_{11}^{(1)}$, $w_{12}^{(1)}$, ..., $w_{32}^{(1)}$ are the weights of the first layer, $w_{11}^{(2)}$, $w_{12}^{(2)}$ are the weights of the second layer, k_1 and k_2 are the thresholds of the first layer and the second layer respectively, and $f(\cdot)$ is the activation function. According to Fig.1 the output y is [29]

$$y = f(f(\sum_{i=1}^{3} x_i w_{i1}^{(1)} + k_1) w_{11}^{(2)} + f(\sum_{i=1}^{3} x_i w_{i1}^{(2)} + k_1) w_{21}^{(2)} + k_2). \tag{1}$$

In this paper, according to the structure of the full connection neural network model, by adjusting the two-layer full connection neural network with eight inputs and eight outputs, the FCLNN as shown in Fig. 2 is designed. Compared with the fully connected neural network, the FCLNN has the following adjustments.

(1) The inputs p_{i1}, p_{i2}, ..., p_{i8} of the input layer are the pixel value of the element of the image, and the pixel values of each input image are recorded as p_i , the input elements of the input layer can be expressed as:

$$p_i = p_{i1}, p_{i2}, p_{i3}, ..., p_{i8}, \tag{2}$$

where $i = 1, 2, 3, ..., \frac{M \times N}{8}$. In Eq. 2, M and N are the number of rows and columns of the image respectively.

(2) The weights from input layer to output layer w_{11}, w_{12}, w_{13}, ..., w_{84}, which is updated by a matrix with the size of 8×8 and each row is randomly arranged from 0 to 7 and the pseudo-random sequence generated by the chaotic systems.

(3) The activation function in neurons (yellow ball in Fig. 2 is replaced by a novel S-box proposed by Lu Qing et al. [30]. The S-box has very small linear probability and differential probability values, and has a good nonlinear average value. The S-box is suitable for cryptographic system. Table.1 shows the new S-box.

(4) The output layer has eight outputs, and the outputs of the output layer are the element value processed by weights and activation function. The output elements of the output layer can be expressed as:

$$c_i = c_{i1}, c_{i2}, c_{i3}, ..., c_{i8}, \tag{3}$$

where $i = 1, 2, 3, ..., \frac{M \times N}{8}$.

This structure, in Fig.2, is applied to image encryption as a part of the algorithm proposed in this paper. The new network inputs eight pixels of gray image each time and outputs eight pixels processed by the network.

3 The proposed scheme

This paper proposes an image encryption algorithm based on FCLNN and edge pixel reset. The structure of the algorithm is shown in Fig. 3. The encryption algorithm is divided into four parts: the image edge pixel reset, CST, FCLNN and bidirectional diffusion. The above encryption part will be described in detail in the following section. In order to enhance the encryption effect of the algorithm, the last three parts of the

Table 1 The new S-box

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	114	75	39	161	61	14	225	150	180	126	232	155	171	129	143	26
2	186	76	234	247	53	185	187	227	106	192	99	31	94	215	219	20
3	110	105	112	60	52	90	188	221	8	48	208	107	201	24	212	19
4	49	191	91	138	97	238	140	220	122	63	139	146	167	137	28	88
5	135	4	222	18	36	168	181	32	9	117	83	148	190	127	102	236
6	205	82	121	199	252	147	67	133	204	111	98	210	173	243	1	184
7	174	230	59	30	176	21	160	62	202	145	195	209	119	96	45	141
8	245	44	78	29	43	177	12	194	156	38	151	50	213	244	22	142
9	170	226	101	72	152	115	217	2	163	109	239	37	104	196	3	189
10	198	218	57	124	27	134	175	74	87	108	89	224	125	237	65	118
11	197	5	158	66	42	157	229	255	211	207	55	203	169	123	56	149
12	242	254	200	100	95	69	46	23	40	251	7	6	103	216	178	79
13	240	253	131	15	183	113	246	93	71	153	249	77	248	10	172	250
14	35	41	132	25	33	47	223	86	81	154	136	233	13	68	64	54
15	166	120	84	17	193	214	0	85	73	92	70	164	182	16	206	130
16	144	228	11	179	80	159	116	128	235	51	241	165	34	231	162	58

algorithm are used to perform n rounds of image processing. On the basis of ensuring security and efficiency, we set n to 2 in this paper. If you want a higher level of security, you can set n to a larger integer.

3.1 Image edge pixel reset

In this paper, we use two chaotic systems: logistic [31] and Fractional-order discrete Hopfield neural network (FODHNN) [13]. The definitions of logistic and FODHNN Chaotic systems are:

$$a_{i+1} = \mu a_i(1 - a_i), \tag{4}$$

and

$$\begin{cases} x_{i+1} = x_i + \frac{h^v}{\Gamma(1+v)}(-x_i + 2\sin(x_i) + \sin(y_i) - 9\sin(z_i)), \\ y_{i+1} = y_i + \frac{h^v}{\Gamma(1+v)}(-y_i - 9\sin(x_i) + 2\sin(y_i) + \sin(z_i)), \\ z_{i+1} = z_i + \frac{h^v}{\Gamma(1+v)}(-z_i + \sin(x_i) - 9\sin(y_i) + 2\sin(z_i)), \end{cases} \tag{5}$$

respectively; where $\mu \in [0,1]$, $i \in N_0$, $h \in R_+$ denotes the discretization step size, v is the fractional-order.

In grayscale image, the image information contained in the lowest bit-plane is very small [32]. Table. 2 shows the percentage of image information contained in different bit-planes. In order to enhance the ability of the proposed algorithm to resist various types of attacks, we design a plaintext-related key generation method by the image edge pixel reset algorithm. Different from some previous algorithms that directly use the hash code of ordinary image as the key, we use random noise to reset the least-significant bit of the edge pixels of the image. Although this operation changes the information of the original image, the information content of the lowest bit of the image is very small, and we only reset least-significant bits of the edge pixels, so the image after reseting is visually consistent with the original image. Fig. 4 shows how to reset the edge pixels value of "Boat", where Fig. 4(a) is the original image of the "Boat", Fig. 4(b) is the highest bit-plane image of the "Boat", Fig. 4(c) is the lowest bit-plane image of the "Boat", and Fig. 4(d) is the "Boat" after reseting least-significant bits of the edge pixels. The detailed steps of this part can be described as follows:

Table 2 Information percentage in different bit-planes

Bit-plane	Percentage(%)	Bit-plane	Percentage(%)
1	0.39	5	6.27
2	0.78	6	12.55
3	1.57	7	25.10
4	3.14	8	50.20

Step1: Input the plain gray image P with the size of $M \times N$, divide the gray image into eight bit-planes, use random noise to reset the least-significant bits of the edge pixels of the image, and then recombine the eight bit-planes into ordinary gray image P'.

Step2: Calculate a_0 from image P', use a_0 as the initial value of logistic system (4).

$$a_0 = \frac{\sum P'(i,j)}{M \times N \times 255}, \tag{6}$$

where $P'(i,j)$ represents the pixel value at position (i,j) of the image. The parameter μ is set to 3.99, and the logistic system is iterated for $M \times N$ times to obtain the chaotic sequence A.

Step3: The image P' is convolved with chaotic sequence A, and a plaintext-related matrix KM with size of 8×8 is obtained.

$$KM = P' * A, \tag{7}$$

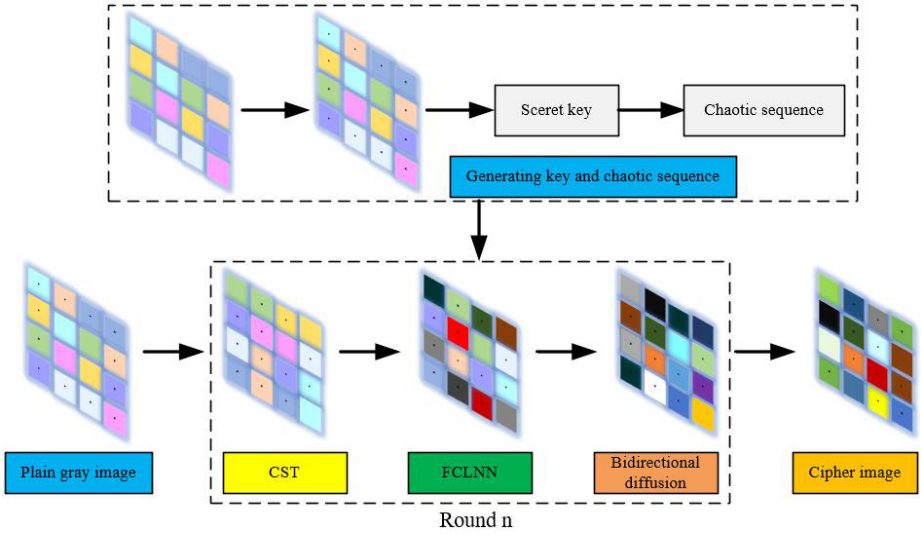

Fig. 3 Structure of the proposed algorithm

where $*$ denotes convolution operation.

Step4: This algorithm uses different keys each round. Six rows are randomly selected from the KM matrix to generate the keys, shown as in Eq. 8. This will introduce more randomness and expand the key space. Here we randomly choose six lines, and the keys generation formula is defined as:

$$\begin{cases} m_1 = mod(sum(KM(l_1:,) + KM(l_2:,)), 256)/256, \\ m_2 = mod(sum(KM(l_3:,) + KM(l_4:,)), 256)/256, \\ m_3 = mod(sum(KM(l_5:,) + KM(l_6:,)), 256)/256. \end{cases} \tag{8}$$

where $mod(\cdot)$ represents the modular operation, $sum(\cdot)$ is the sum of all the elements. $l_1, l_2, l_3, l_4, l_5, l_6$ are randomly selected row numbers.

Step5: The initial values of the FODHNN system (5) are obtained by the generated keys, the initial values are given by:

$$\begin{cases} x_1 = k_1 + m_1, \\ y_1 = k_2 + m_2, \\ z_1 = K_3 + m_3. \end{cases} \tag{9}$$

where k_1, k_2, k_3 are the given values, and then iterate FODHNN system (5) $M \times N + 1000$ times, and discard the first 1000 values to minimize the impact of the transient, and finally the chaotic sequences X, Y, Z are obtained.

We use edge pixel reset algorithm to reset the least-significant bits of the edge pixels of the image and compare the a_0 value of the image without noise. The two a_0 values change in the sixth place after the decimal point. Although the change is very small, because of the sensitivity of the chaotic systems to the initial value, the generated chaotic sequence is completely different. To a certain extent, it can resist the attacker to analyze the keys through the image information. Fig.5 shows the result of reseting the least-significant bits of the edge pixels of the image in the encryption process. Since the noise is random, the images after reseting the edge pixels are different. So each encryption will generate a unique cipher image. When two users encrypt the same image, they cannot decrypt each other. This operation enhances the ability to resist various attacks.

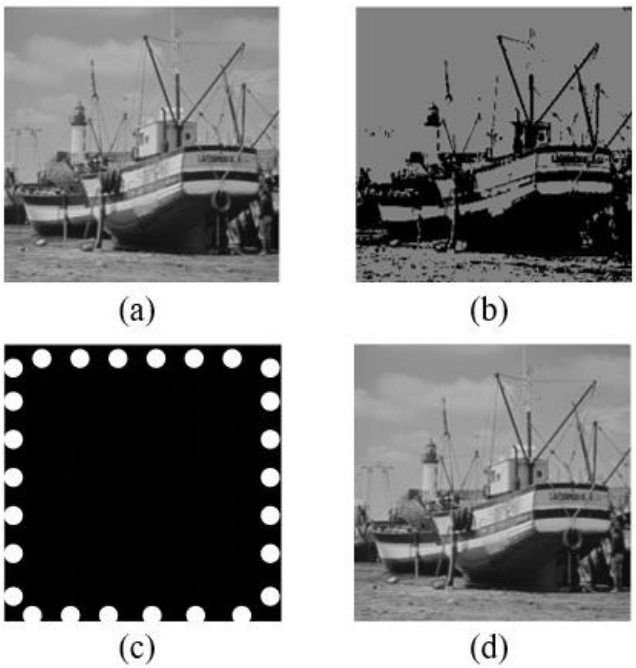

Fig. 4 Comparison before and after noise reset: the white balls in (c) are the reseted pixels

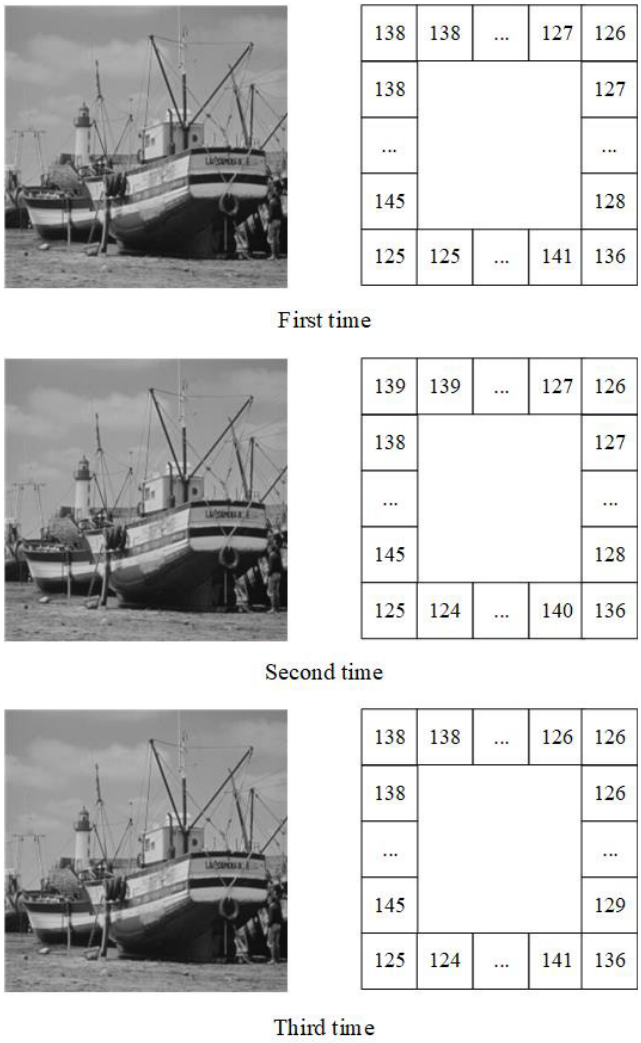

Fig. 5 Reset the least-significant bits of the edge pixels of the image

3.2 Cyclic shift transformation (CST)

In this section, a method of CST is proposed to scramble the pixel position of the image, so as to reduce the correlation between pixels. Some existing algorithms only exchange the position of rows or columns to achieve the purpose of pixel position scrambling, which may lead to the decrease of encryption performance and efficiency. In order to overcome these shortcomings, this paper uses the row and column cyclic shift method based on chaotic sequence. The index sequence of chaotic sequence is used to transform rows and columns. The step size of the shift is the value of the corresponding index sequence. If the corresponding value is even, the row and column are shifted to the right and down, respectively. Similarly, if the corresponding value is odd, the rows and columns are shifted to the left and up, respectively. The detailed steps of this part can be described as follows:

Step1: The sequences M_1 and N_1 of length M and N are intercepted from chaotic sequence X respectively, and t_1 and t_2 are the starting points of intercepting. Here, we set the values of t_1 and t_2 to 2000 and 3000 respectively.

$$\begin{cases} M_1 = X(t_1 : t_1 + M - 1), \\ N_1 = X(t_2 : t_2 + N - 1). \end{cases} \tag{10}$$

Step2: Sort M_1 and N_1 from small to large, and get two index sequences MM and NN.

$$\begin{cases} MM = Sort(M_1), \\ NN = Sort(N_1). \end{cases} \tag{11}$$

Step3: According to the sequence MM, the row of the image P' is circularly shifted, and the shift step is the value of the corresponding element of the sequence MM. If the ith value of the sequence MM is even p, then the ith row of the image P' circularly shifts to the right by p cells; if the ith value of the sequence MM is odd p, then the ith row of the image P' circularly shifts to the left by p cells. After the cyclic shift of the row, to obtain the result as Q.

Step4: Similarly, the columns of the image Q are circularly shifted according to the sequence NN. If the jth value of sequence NN is an even number q, then the jth column of image Q circularly shifts to the down by q cells; if the jth value of sequence NN is an odd number q, then the jth column of image Q circularly shifts to the up by q cells. After the cyclic shift of the column, to obtain the result as P.

The process of Step 3 and Step 4 is described in Algorithm 1.

For simplicity, we provide a numerical example to explain its detailed operation on a 4×4 image. Fig. 6 shows the process of CST.

3.3 Bit-level image diffusion based on FCLNN

In this section, a method of FCLNN is proposed for bit-level image diffusion. Different from the previous algorithm, the proposed image encryption system uses FCLNN to realize the bit-level diffusion of image information. The elements of the image matrix are input into the network, and the weights of the network are generated by a random matrix and chaotic sequence. Then the improved S-box is used as the activation function, and the diffused image matrix is output. The detailed steps of this part can be described as follows:

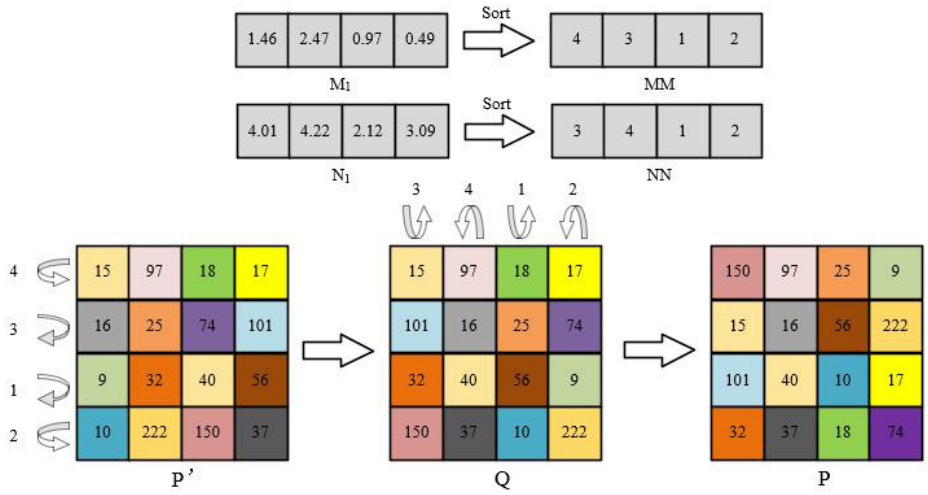

Fig. 6 The process of image CST

Algorithm 1 The process of Step 3 and Step 4

Input: Image matrix P', chaotic sequence MM, NN.
1: **for** $i = 1$ to M **do**
2: **if** mod $(p,2) = 0$ **then**
3: the ith row of the image P' circularly shifts to the right by p cells
4: **else**
5: the ith row of the image P' circularly shifts to the left by p cells
6: **end if**
7: **end for**
8: Get the image Q
9: **for** $j = 1$ to N **do**
10: **if** mod $(q,2) = 0$ **then**
11: the jth row of the image Q circularly shifts to the down by q cells
12: **else**
13: the jth row of the image Q circularly shifts to the up by q cells
14: **end if**
15: **end for**
Output: the scrambling image matrix P

Step1: Convert the image matrix P to 8 rows and $M \times N/8$ columns.

$$P = Reshape(P, 8, M \times N/8). \tag{12}$$

Step2: The chaotic sequence X is converted to integer number between 0 and 7, and then convert to 8 rows and $M \times N/8$ columns matrix XR.

$$\begin{cases} XR = mod(Floor(X \times 10^14), 8), \\ XR = Reshape(XR, 8, M \times N/8). \end{cases} \tag{13}$$

where Floor$(\cdot)$ represents the maximum integer not greater than the given value. Step3: Generate a matrix W whose size is 8×8 and each row is randomly arranged from 1 to 8. Algorithm 2 shows the pseudo code generated by matrix W. Step4: Input the image matrix P into

Algorithm 2 The process of matrix W generation

1: **for** $i = 1$ to 8 **do**
2: $W(i, :)$ = randperm (8);
3: **end for**
Output: matrix W

FCLNN, convert the elements of image P into 8-bit binary form, and calculate one row of image P each time. Each row of matrix W is circularly shifted to the left by using the corresponding row of chaotic sequence XR, and the shift step is the value of sequence XR. The shifted matrix W is used as the weights. The number of the corresponding position of the image P is selected to the output element by the value of the weight. Convert the output elements to decimal representation, and then use the activation function S-box to process to get the diffused elements. From the first row to the last row of the image matrix P, we process FCLNN on each row, and finally get the diffused image matrix T. Algorithm 3 shows the pseudo code of FCLNN bit-level diffusion process. For simplicity, we provide an example to illustrate its detailed operation on the first two lines of a 4×4 image. Fig. 7 shows the FCLNN bit-level diffusion process, here we only describe the diffusion process of the first two rows elements, and use the first two rows of elements $W(1, :)$ and $W(2, :)$ of matrix W.

3.4 Bidirectional diffusion

Encryption algorithm must have diffusion property. Some algorithms use chaotic sequences to XOR with image pixels directly to achieve diffusion, which may provide useful information to attackers for cryptanalysis. We designed a bidirectional diffusion method for image, namely forward diffusion and reverse diffusion. Fig. 8

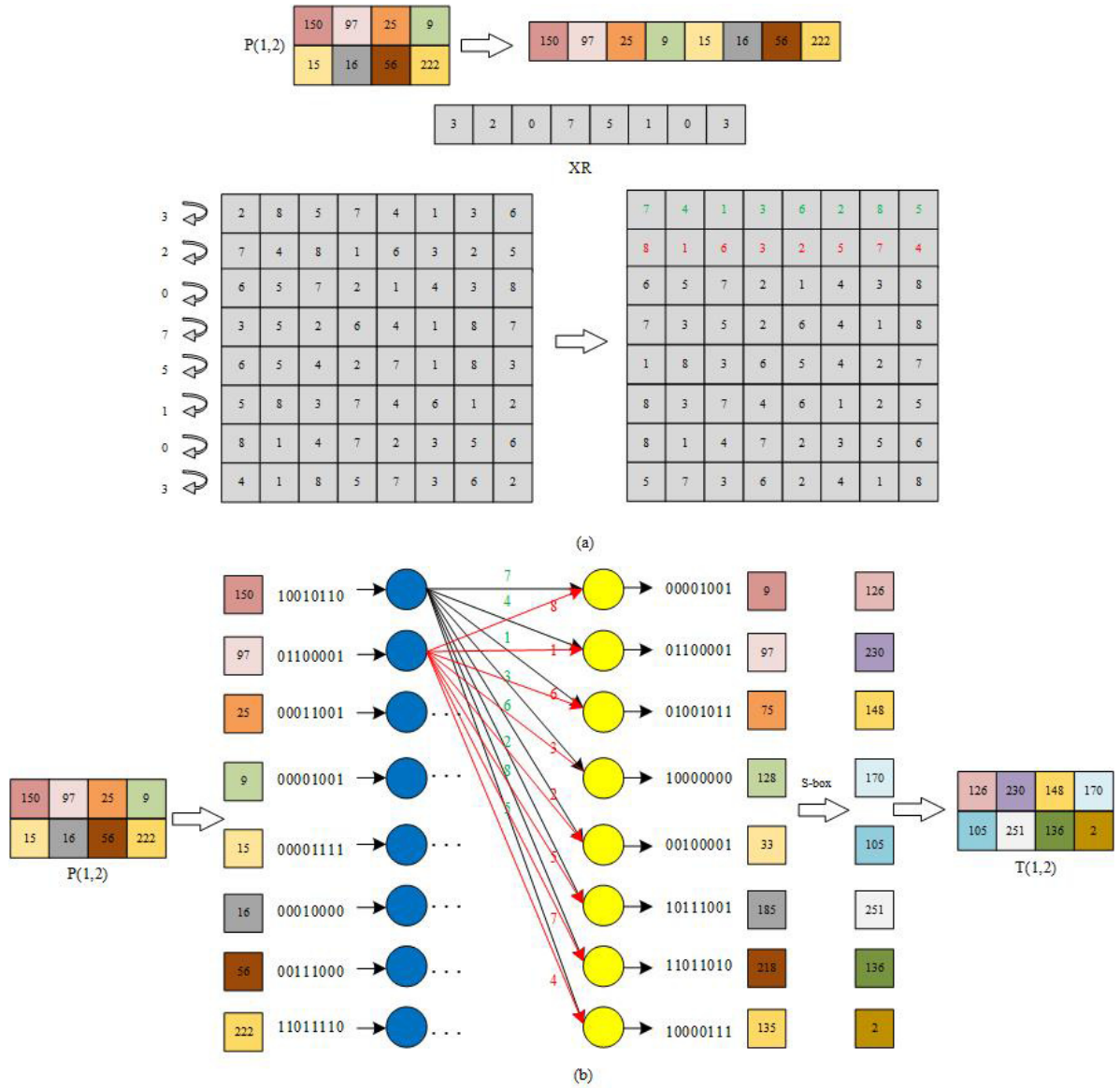

Fig. 7 FCLNN: (a) generating matrix W (b) Bit-level diffusion process of FCLNN

Algorithm 3 The FCLNN bit-level diffusion process

Input: Image matrix P, chaotic sequence XR, matrix W.
The image P is transformed into 8-bit binary form
for $i = 1$ to $M \times N/8$ **do**
  for $j = 1$ to 8 **do**
    y = bitget ($P(i, j)$, 8: -1:1)
    a = $W(j, :)$
    z = $[a(XR(i, j) + 1 : end) \quad aa(1 : XR(i, j))]$
    m = 1 to 8
    t1(m) = y(z(m))
    t2(:, j) = t1
  end for
  t3 = convert to decimal representation (t2)
  t3 = new S-box(t3)
  T(i, :) = t3
end for
Output: the diffused image matrix T

shows the bidirectional diffusion process. In this way, the change of each pixel can affect all the pixels in the image. The detailed steps of this part can be described as follows:

Step1: The chaotic sequences Y and Z are converted to integer number between 0 and 255 respectively. The results are YR and ZR.

$$\begin{cases} YR = mod(Floor(Y \times 10^{14}), 256), \\ ZR = mod(Floor(Z \times 10^{14}), 256). \end{cases} \tag{14}$$

Step2: Starting from the upper left corner of the image, from left to right, top to bottom. Diffusion the

image T to obtain the middle image D.

$$D_{i,j} = \begin{cases} T_{i,j} \bigoplus T_{M,N} \bigoplus YR_{i,j}, \; if \; i = 1, j \neq 1, \\ T_{i,j} \bigoplus T_{i,j-1} \bigoplus YR_{i,j}, \; if \; j \neq 1, \\ T_{i,j} \bigoplus T_{i-1,N} \bigoplus YR_{i,j}, \; if \; i \neq 1, j = 1. \end{cases} \tag{15}$$

Step3: Start at the bottom right corner of the image, right to left, bottom to top. The image D is diffused to obtain the encrypted image C.

$$C_{i,j} = \begin{cases} D_{i,j} \bigoplus D_{1,1} \bigoplus ZR_{i,j}, \; if \; i = M, j \neq N, \\ D_{i,j} \bigoplus C_{i,j+1} \bigoplus ZR_{i,j}, \; if \; j \neq N, \\ D_{i,j} \bigoplus C_{i+1,1} \bigoplus ZR_{i,j}, \; if \; i \neq M, j = N. \end{cases} \tag{16}$$

3.5 Decryption process

The image encryption algorithm proposed in this paper is symmetric encryption algorithm, so decryption is the reverse process of encryption. Firstly, the cipher image is input into the "Inverse of Bidirectional diffusion" module, and then the obtained sequence is input into the "Inverse of FCLNN" module. After that, the output of the previous step goes through the "Inverse of CST " module. After n rounds of decryption process, finally, the obtained sequence is reshaped into a matrix of size $M \times N$, that is, the decrypted image. Fig. 9 shows the decryption process

4 Experiment and analysis of encryption scheme

In this section, we use MATLAB R2015b platform to test the performance of our proposed encryption method. The computer hardware parameters used are Microsoft Windows 10 home (64 bit), Intel Core i5-9300H@2.40 GHz and 8GB DDR4@2666 MHz memory. The proposed algorithm can encrypt different types of images, so we choose "Baboon", "Peppers", all black, and all white images as our test images. For easy comparison, all the test images are 8-bit gray images, the size are 256×256, and the encryption rounds are $n = 2$. User provided values of k_1,k_2,k_3,h and v are (0.07, 0.8, -6.2, 0.05, 0.6). The performance of the proposed image encryption algorithm is evaluated from different aspects, and the results are compared with the existing methods to prove the effectiveness and security of our image encryption scheme. Fig. 10 shows the results of encryption and decryption using this algorithm, which illustrates that it can effectively eliminate the intuitive visual information about the original image and can completely reconstruct this information.

Table 3 Key space

Algorithm	Key space
Proposed	2^{749}
Ref. [33]	2^{256}
Ref. [34]	2^{446}
Ref. [35]	2^{185}
Ref. [36]	2^{283}

4.1 Key space

The key space is the range of the key size of the encryption algorithm. The encryption algorithm should have enough key space to resist violent attacks. Generally, the key space is designed to be greater than 2^{100} [5]. In this algorithm, the key is composed of a five-parameter vector provided by the user and a 512-bit matrix W. The precision of user's input data is 10^{15}, and the fractional order and discrete step are also 10^{15}. In addition, the key space caused by matrix W is 2^{512}. Therefore, our key space is about $10^{75} \times 2^{512} \approx 2^{749}$. The results are shown in Table 3. From the table, we can see that the key space is larger than the minimum of 2^{100}, which shows that this algorithm has the ability to resist violent attacks. At the same time, it has a larger key space than the previous algorithms.

4.2 Key sensitivity analysis

Encryption algorithm must be very sensitive to key. Key sensitivity means that if the key changes slightly during encryption and decryption, it will produce completely different encryption and decryption results [3]. Here, we tested the sensitivity of the encryption key and decryption key to verify the key sensitivity of the proposed algorithm. For the sake of brevity of the article, only the experimental results of changing the values of k_1,k_2, and k_3 input by the user are described. The keys used in the experiment are as follows:

$$\begin{cases} K_1 = (0.07, 0.8, -6.2, 0.05, 0.6), \\ K_2 = (0.07 + 10^{-15}, 0.8, -6.2, 0.05, 0.6), \\ K_3 = (0.07, 0.8 + 10^{-15}, -6.2, 0.05, 0.6), \\ K_4 = (0.07, 0.8, -6.2 + 10^{-15}, 0.05, 0.6). \end{cases} \tag{17}$$

Fig. 11 shows the key sensitivity analysis results of the proposed algorithm in the "Baboon" encryption process. In Fig.11, b-e are the encryption results using K_1,K_2,K_3 and K_4 respectively, and f-h are the differences between the encryption results using K_1 and those using K_2,K_3 and K_4 respectively. It can be observed that the keys are only slightly changed, but the results of encryption are completely different. Fig. 12 shows the experimental results of the key sensitivity

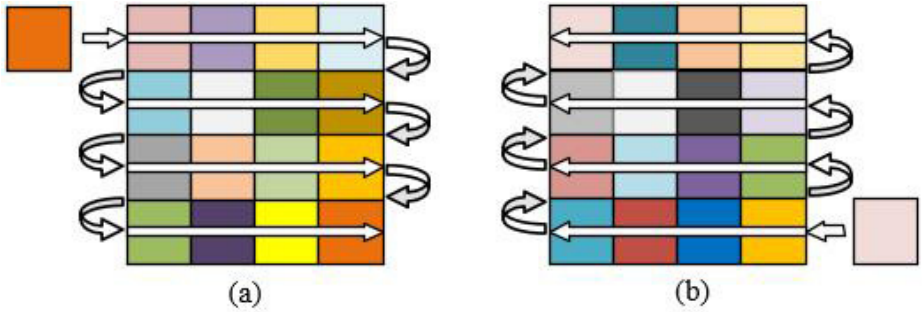

Fig. 8 Bidirectional diffusion: (a) forward diffusion (b) reverse diffusion

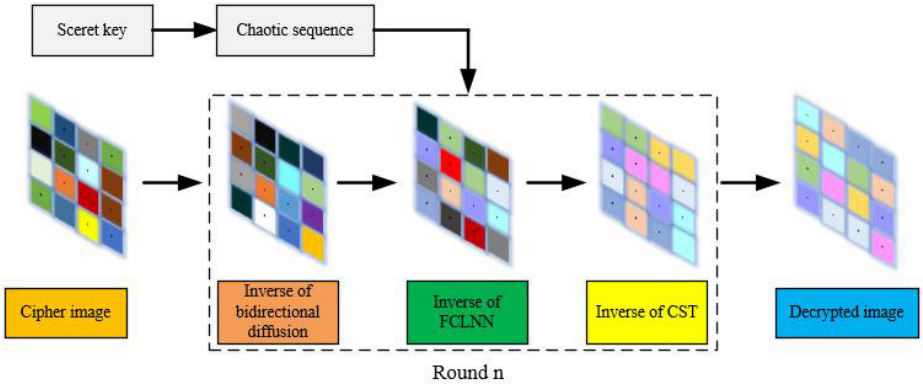

Fig. 9 Structure of decryption process

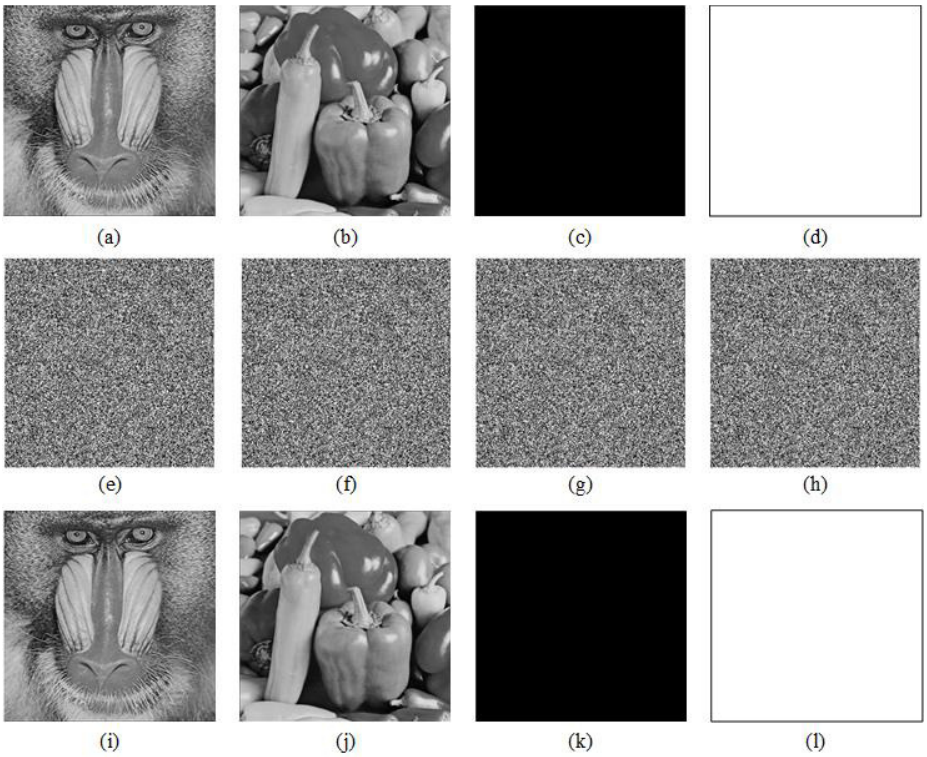

Fig. 10 Encryption and decryption results: (a) Baboon; (b) Peppers; (c) All black; (d) All white; (e)-(h) cipher images of (a)-(d), respectively; (i)-(l) decryption images of (e)-(h), respectively

of the proposed algorithm in the decryption process of "Baboon" cipher image, one can see that only the correct key can accurately restore the original image. The correct decryption result cannot be obtained if the key differs from the correct key by only 10^{-15}. The experimental results show that the keys of our algorithm are very sensitive.

4.3 Histogram analysis

The histogram of a digital image can count the number of pixels with the same pixel value and can directly reflect the distribution of all pixel values in the image. Usually, the pixel value distribution of a meaningful image is uneven, so its histogram is fluctuating. In contrast, the pixel value distribution of the encrypted image is uniform, and its histogram is relatively flat [37]. Fig. 13 shows the histograms of plain images "Baboon", "Peppers ", all black and all white images (Fig. 10 a-d), and Fig. 14 shows the histograms of their corresponding cipher image (Fig. 10 e-h). It can be seen that the histograms of the cipher images are quite flat.

The sample standard deviation is used to quantitatively analyze the difference between the histograms of the plain image and the cipher image. For the sample space X_i , i=0,1,. . . ,255, the formula for calculating the sample standard deviation is [38]:

$$s = \sqrt{\frac{1}{256}\sum_{i=0}^{255} X_i^2 - (E(X))^2} \tag{18}$$

where for images of size 256×256, the $E(X)$=256, X_i is the number of times a pixel with a value of i appears in the image, and s represents the sample standard deviation of the histogram. Table 4 lists the sample standard deviations of the histograms of the plain image and the cipher image in Fig. 13 and Fig. 14. It can be seen from the table that the sample standard deviation of the cipher image histogram is about 16, which is far less than that of the plain image histogram. Therefore, the histogram of the cipher image generated by our algorithm is very flat compared with that of the corresponding plain image. As can be seen from both the histogram and the sample standard deviation, our encryption algorithm can hide the distribution information of image pixel value well and can effectively resist statistical analysis.

4.4 X^2 test

In order to quantitatively verify that the pixel values of the cipher image are evenly distributed, X^2 test is performed on the cipher image. Assuming that the distribution of 8-bit gray image with size of $M \times N$ is uniform, the statistical formula is [39]:

$$X^2 = \sum_{i=0}^{255} \frac{(f_i - f_g)^2}{f_g} \tag{19}$$

where i is the pixel value of the image, f_i represents the number of times the pixel with the value of i appears in the image, and f_g=$(M \times N)/256$ represents the average number of occurrences of all pixel values. Table 5 lists the X^2 values of plain image and cipher image. From the table, we can see that the X^2 value of the cipher image is less than 293.2478. In other words, all the cipher images have passed the X^2 test, so it can be proved that the distribution of the pixel values of the cipher image generated by our proposed encryption algorithm is uniform.

Table 4 Sample standard deviation of the histogram

Image	Plain image	Cipher image
Baboon	222.09	15.41
Peppers	178.29	15.67
All black	4087.99	15.58
All white	4087.99	16.05

Table 5 X^2 test results

Image	Plain image	Cipher image
Baboon	49323	219.69
Peppers	31787	240.57
All black	16711680	247.88
All white	16711680	261.41

4.5 Correlation analysis

Meaningful images have a lot of redundant information, and there is a strong correlation between their adjacent pixels. On the contrary, cipher images have no redundant information, and their adjacent pixels have no correlation. In order to prevent attackers from finding a way to crack the encryption algorithm by analyzing the correlation between adjacent pixels, image encryption algorithm should break this high correlation.

To test the image correlation, we randomly select 5000 pairs of adjacent pixels from the horizontal, vertical and diagonal directions of "Baboon" in Fig. 9 and its cipher image, respectively. Fig. 14 shows the correlation of adjacent pixels of the two images in three directions. We can see that the pixels in the three directions of the plain image are distributed near y = x, while the pixels in the three directions of the cipher image are evenly distributed. The results represent that

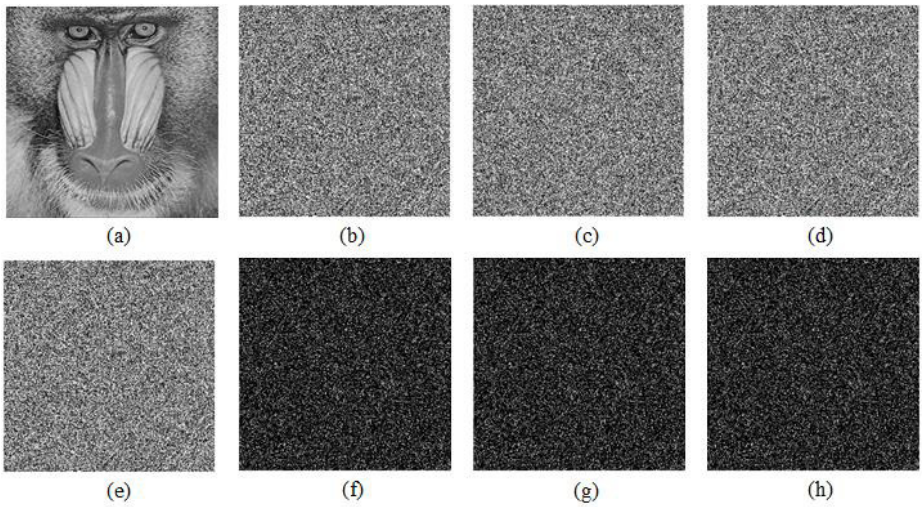

Fig. 11 Key sensitivity analysis in the encryption process : (a)Baboon; (b) cipher image using K_1; (c) cipher image using K_2; (d) cipher image using K_3; (e) cipher image using K_4; (f) difference between (b) and (f), |(b)-(f)|; (g) difference between (b) and (g),|(b)- (g)|; (h) difference between (b) and (h),|(b)-(h)|

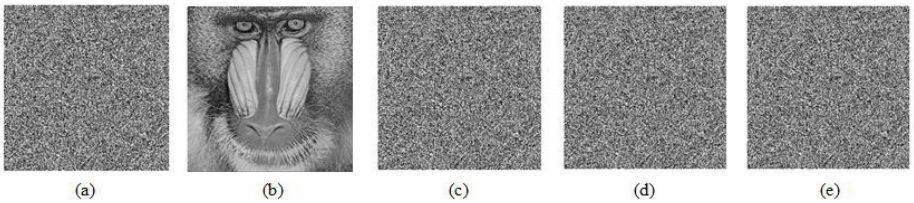

Fig. 12 Key sensitivity analysis in the decryption process: (a) cipher image using K_1; (b) decrypted image using K_1 (c) decrypted image using K_2; (d) decrypted image using K_3; (e) decrypted image using K_4

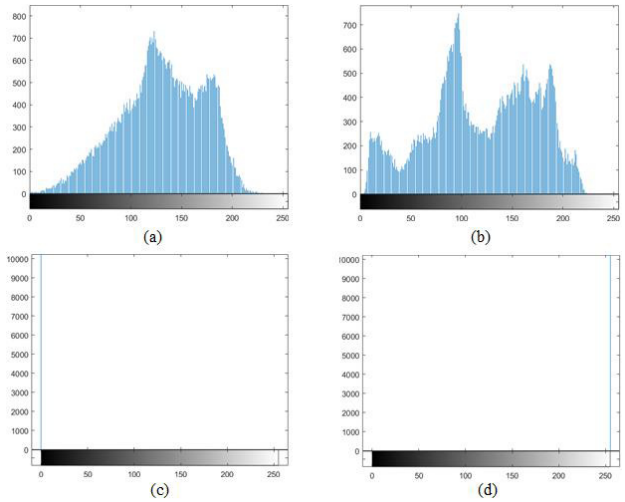

Fig. 13 Histograms: (a)–(d) Histograms of Fig. 10 (a)–(d), respectively

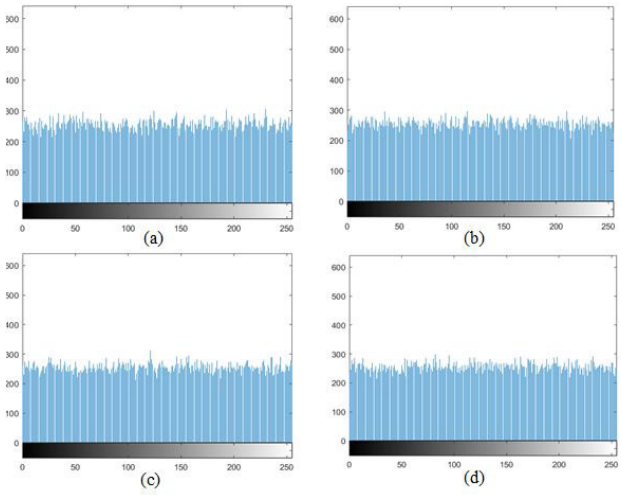

Fig. 14 Histograms: (a)–(d) Histograms of Fig. 10 (e)–(h), respectively

our algorithm breaks the high correlation between the adjacent pixels of the image. At the same time, quantitative methods can also be used to analyze the correlation of images. The following formulas are used to calculate the correlation coefficients of adjacent pixels in the three directions of the plain image and cipher

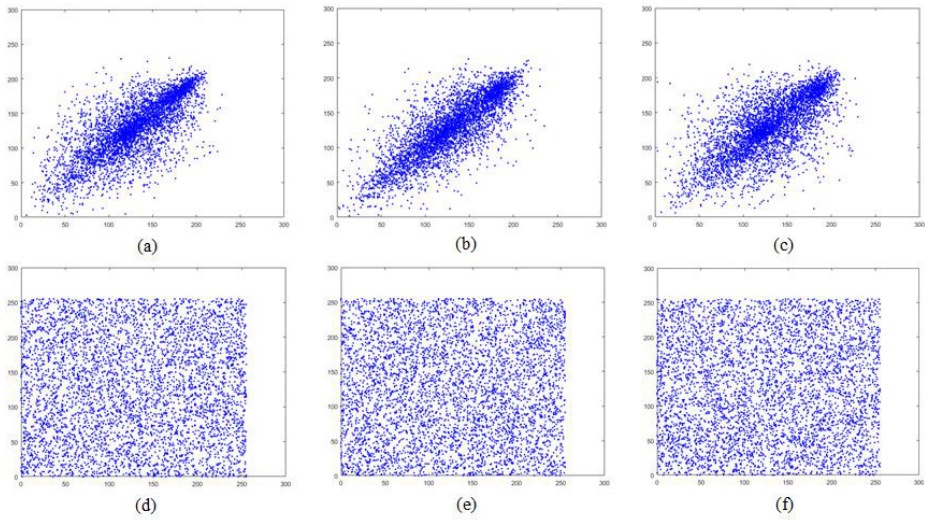

Fig. 15 Adjacent pixel correlation: (a) Horizontal direction of Baboon; (b) Vertical direction of Baboon; (c) Diagonal direction of Baboon; (d) Horizontal direction of Encrypted Baboon; (e) Vertical direction of Encrypted Baboon; (f) Diagonal direction of Encrypted Baboon

image [40].

$$\begin{cases} r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}}, \\ cov(x,y) = \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))(y_i - E(y)), \\ D(x) = \frac{1}{S}\sum_{i=1}^{S}(x_i - E(x))^2, \\ E(x) = \frac{1}{S}\sum_{i=1}^{S} x_i. \end{cases} \tag{20}$$

where x and y are two adjacent sequences of pixels, $cov(x,y)$ is the covariance of x and y, and r_{xy} is the correlation coefficient. r_{xy} is used to measure the correlation of adjacent pixels. If r_{xy} is close to 1, the correlation between adjacent pixels is very high. On the contrary, if r_{xy} is close to 0, the correlation between adjacent pixels is very small.

Table 6 lists the correlation coefficients of adjacent pixels selected from the plain image (Fig. 10a-d) and cipher image (Fig. 10e-h) in the horizontal, vertical and diagonal directions. The correlation coefficient of the plain image in each direction is close to 1, which indicates that the adjacent pixels in the plain image have a strong correlation. However, the correlation coefficient of the cipher image in each direction is very close to 0, that is, the correlation between adjacent pixels in the cipher image is very low. Table 7 compares the correlation coefficient calculation results of "Baboon" with other algorithms. The proposed encryption algorithm successfully destroys the correlation of adjacent pixels, and the algorithm has better resistance to statistical attacks.

4.6 Information entropy analysis

Information entropy reflects the uncertainty of image information, which can be used as an important feature to detect the randomness of image pixel value distribution. The calculation formula of image information entropy is [43]

$$H = -\sum_{i=0}^{255} P_i log_2 P_i \tag{21}$$

where P_i is the frequency of occurrence of the pixel with the value i. When the frequency of each pixel is equal, we can calculate the highest value of information entropy theoretically. $H = -log_2(1/256) = 8$.

The calculation results of information entropy of cipher image (Fig. 10e-h) are shown in Table 8. The results illustrate that the information entropy of the cipher image generated by our algorithm is close to 8, which means that that the pixel values of the cipher image are evenly distributed, and proves that our algorithm has the ability to resist statistical attacks. We compare the results of "Baboon" with other algorithms, and the results are listed in Table 9. The table shows that our algorithm has higher information entropy than other algorithms, which shows that our algorithm has higher security.

4.7 Peak signal to noise ratio (PSNR)

PSNR is a measure of image quality. Here, we first use PSNR to analyze the difference between the plain image and the cipher image. Second, in order to verify that

Table 6 Correlation coefficients of images

Image	Plain image			Cipher image		
	Horizontal	Vertica	Diagonal	Horizontal	Vertical	Diagonal
Baboon	0.9631	0.9292	0.9089	0.0061	-0.0006	-0.0005
Peppers	0.9680	0.9654	0.9443	-0.0032	-0.0028	-0.0006
All black	1.0000	1.0000	1.0000	0.0028	-0.0055	-0.0043
All white	1.0000	1.0000	1.0000	0.0034	-0.0045	0.0204

Table 7 Comparison of different algorithms for Baboon (256*256)

Algorithm	Horizontal	Vertica	Diagonal
Proposed	0.0061	-0.0006	0.0005
Ref. [41]	-0.0159	0.0021	-0.0017
Ref. [42]	0.0057	0.0087	0.0037
Ref. [23]	-0.0061	0.0032	0.0105
Ref. [36]	-0.0018	0.0007	-0.0013

Table 8 Information entropy of cipher images

Cipher image	Information entropy
Baboon	7.9977
Peppers	7.9974
All black	7.9977
All white	7.9975

Table 9 Comparison of different algorithms for Baboon (256*256)

Algorithm	Information entropy
Proposed	7.9977
Ref. [40]	7.9971
Ref. [23]	7.9976
Ref. [38]	7.9973
Ref. [6]	7.9970

our algorithm can generate completely different cipher images when encrypting the same image multiple times, we use PSNR to analyze multiple cipher images of the same image. Finally, due to the use of random noise in the encryption process to reset least-significant bits of the edge pixels, so our algorithm has a loss. In order to measure the decryption result of the algorithm, we use PSNR to analyze the difference between the original image and the decrypted image. The formula is as follows [21]:

$$\begin{cases} MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(P(i,j)-C(i,j))^2}{M\times N}, \\ PSNR = 10\times log_{10}(\frac{max^2}{MSE}), \end{cases} \tag{22}$$

where $M \times N$ is the size of the image, $P(i,j)$ and $C(i,j)$ are the pixel value of the image, MSE is the mean square error of image P and C, and max is the maximum pixel value of plain image.

Table 10 PSNR of plain image and cipher image

Image	PSNR
Baboon	9.5744
Peppers	8.9371
All black	4.7648
All white	4.7586

The smaller the PSNR value, the greater the difference between images, indicating the better encryption effect of the algorithm. The larger the PSNR value, the better the quality of the reconstructed image. Table 10 lists the PSNR of the plain image and the cipher image in Fig. 10a-d. From the table, as we can see that the PSNR is less than 8. The test results show that our algorithm has good encryption performance. Table 11 lists the PSNR between the cipher images generated by encrypting the plain images in Fig.10a-d three times. It can be seen from the table that the PSNR between cipher images is less than 8. This means that the same image encryption, each time will generate a completely different cipher image. That is, when different users encrypt the same image, even using the same key, they cannot decrypt each other. To ensure that our algorithm can resist all kinds of attacks. Table 12 lists the PSNR values between the original image and the decrypted image in Fig.10. We can see that the PSNR values are all greater than 50, indicating that the loss caused by the algorithm is negligible.

4.8 Chosen-plaintext attacks

Chosen-plaintext attack is a common cryptanalysis technique. By chosen-plaintext attack, the attacker can determine the relationship between plaintext and ciphertext, and then crack the key stream of other images. The proposed algorithm obtains partial keys by convoluting plaintext images and chaotic sequences. This means that when the image information changes, a completely different key will be obtained. Moreover, we insert random noise into the lowest bit of the outermost image. The noise is different each time, so the generated keys are also different. Even for the same image, the keys generated each time are different. These will

Table 11 PSNR of cipher images of the same image

Image	First and second	First and third	Second and third
Baboon	7.7673	7.7553	7.7773
Peppers	7.7528	7.7376	7.7647
All black	7.7531	7.7720	7.7997
All white	7.7237	7.7305	7.7201

The first, second and third are the first, second and third encrypted cipher images respectively

Table 12 PSNR of decrypted image and original image

Image	PSNR
Baboon	53.7299
Peppers	52.2476
All black	52.1922
All white	50.4423

Table 13 Average NPCR and UACI values

Image	NPCR (%)	UACI (%)
Baboon	99.6063	33.4619
Peppers	99.6139	33.4685
All black	99.6002	33.4695
All white	99.6047	33.4701

directly affect the generated chaotic sequence, and then affect all the pixels of the cipher image. At the same time, in the diffusion process, the proposed algorithm will spread any small changes of the pure image to all the pixels of the cipher image. Therefore, the algorithm can resist the chosen-plaintext attack, and the attacker cannot obtain the key of other images from the plaintext attack. If an algorithm can resist chosen-plaintext attack, it can also resist ciphertext-only attacks, known-plaintext attacks and chosen-ciphertext attacks [44].

4.9 Differential attack

Differential attack is often used for cryptanalysis. The attacker changes the value of one or more pixels of the plain image, and finds a way to crack the algorithm by comparing the difference between the new cipher image and the original cipher image.

In order to prove the ability of the proposed algorithm to resist differential attacks, change the pixel value of "Baboon" at any random position (the pixel coordinates in this experiment are (100, 100)) to generate a new "Baboon" image, and then use the same key to encrypt the two images and calculate the difference between the two encryption results. As can be seen from Fig. 16, the two encryption results are different. In order to quantitatively test the ability of the proposed algorithm to resist differential attacks. Use the number of pixel change rate (NPCR) and the unified average changing intensity (UACI) to measure the difference between two images, denoted as [45]:

$$\begin{cases} NPCR = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \\ UACI = \frac{1}{M \times N}[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}] \times 100\%, \end{cases} \tag{23}$$

where $M \times N$ represents the size of the image, C_1 and C_2 are two cipher images with only one different pixel. If $C_1(i,j) \neq C_2(i,j)$, $D(i,j) = 1$; otherwise,$D(i,j) = 0$. The theoretical values of NPCR and UACI are 99.6094% and 33.4635%, respectively.

In the experiment, four images (Fig. 10a-d) are selected as test images, and each image is tested multiple times, and the average value is calculated. The test results are shown in Table 13. It can be seen that NPCR and UACI are very close to their theoretical values, indicating that it is difficult for attackers to conduct cryptanalysis through differential attacks. In addition, we compare the experimental results of "Baboon" with other existing algorithms. As shown by the data in Table 14, our results are closer to the theoretical values of NPCR and UACI. Therefore, this algorithm has a good ability to resist differential attack.

Table 14 Comparison of different algorithms about NPCR and UACI for Baboon (256*256)

Algorithm	NPCR (%)	UACI (%)
Proposed	99.6063	33.4619
Ref. [41]	99.1617	33.3261
Ref. [42]	99.3194	33.3495
Ref. [23]	99.5948	33.4382
Ref. [38]	99.6094	27.9303

4.10 Robustness to data loss and noise

When the cipher image is transmitted in the communication system, the data may be lost or affected by noise due to various reasons. A good encryption algorithm can recover the original image as much as possible from the lost data or the cipher image affected by noise.

In this section, we test the ability of the proposed algorithm to resist data loss and noise. The encrypted "Baboon" image data is cut by 25% and 50%, and then

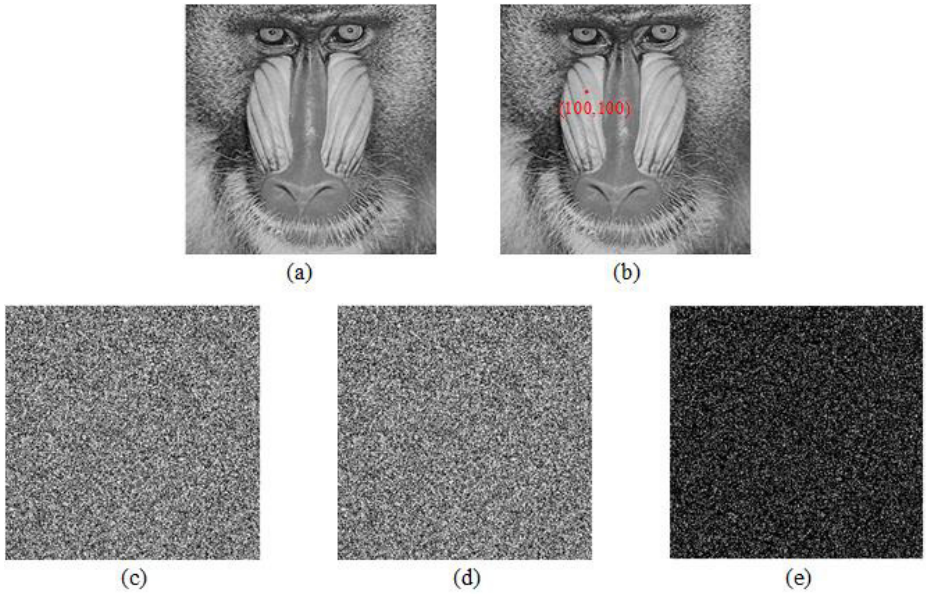

Fig. 16 Differential attacks analysis: (a) Baboon; (b) Change the pixel value at position (100,100); (c) Cipher images of (a); (d) Cipher images of (b); (e) difference between (c) and (d), |(c)-(d)|

the correct key is used to decrypt these processed cipher images. The decrypted image as shown in Fig. 17 restores most information of the original image data.

Salt & Pepper Noise (SPN) and Gaussian Noise (GN) are common noises. Add SPN and GN of different strengths to the "Baboon" cipher image, and then decrypt them with the correct key. From Fig.18, the decrypted images successfully display the information of the original image. This experiment proves that the proposed algorithm has good resistance to data loss and noise.

5 Conclusion

This paper proposes an image encryption algorithm based on FCLNN and edge pixel reset. The algorithm consists of four steps: image edge pixel reset, CST, FCLNN and bidirectional diffusion. In the image edge pixel reset part, using random noise to reset the least-significant bits of the edge pixels of the image, then the key is set by the convolution result of image and chaotic sequence. Each encryption can generate a completely different key, it can enhance the ability of the algorithm to resist various attacks. CST is a cyclic shift of rows and columns on the image, and determine the number of moving steps by chaotic sequence, this step allows the image to be completely scrambled. The FCLNN imitates the structure of fully connected neural network to conduct bit-level diffusion of images. Bidirectional diffusion is the forward and backward diffusion of image. These two parts can spread the tiny change of image pixel value to the whole cipher image, so as to significantly improve the security level of encryption. Simulation results show that the algorithm has negligible loss and large key space, can withstand common attacks such as statistical attack, differential attack, noise attack and data loss attack, and is superior to other algorithms in some aspects. In this paper, we did not test the color image, but the proposed encryption algorithm can encrypt the color image. In the future work, We plan to propose a more complex key generation method, and design an image encryption algorithm based on multi-layer neural network with higher security and efficiency.

Funding

This research was funded by the National key Research and Development projects (2018YFB1800303) the Natural Science Foundation of Jilin Province (20190201188JC), the Research on teaching reform of higher education in Jilin Province (JLLG685520190725093004), and the subject of Educational Science Planning in Jilin Province (GH180148).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

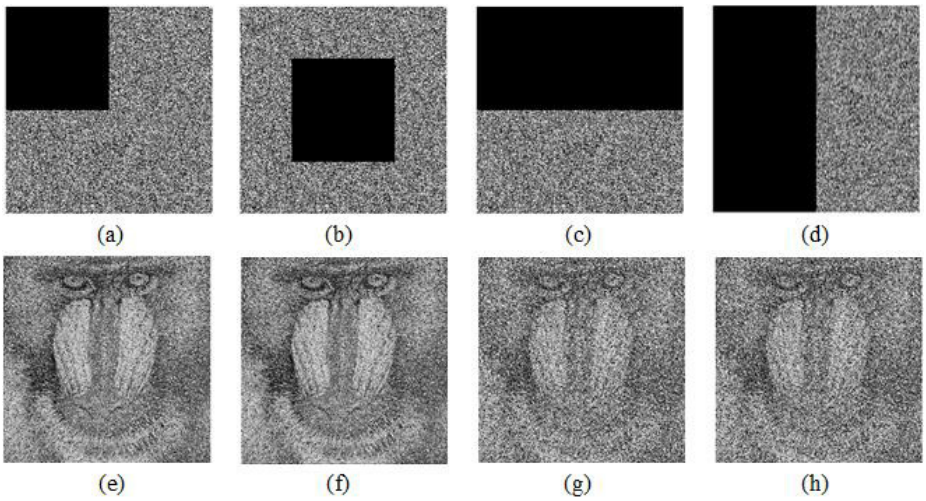

Fig. 17 Data loss analysis: (a) and (b)25% data loss of encrypted Baboon; (c) and (d)50% data loss of encrypted Baboon; (e) decrypted Baboon of (a); (f) decrypted Baboon of (b); (g) decrypted Baboon of(c); (h) decrypted Baboon of (d)

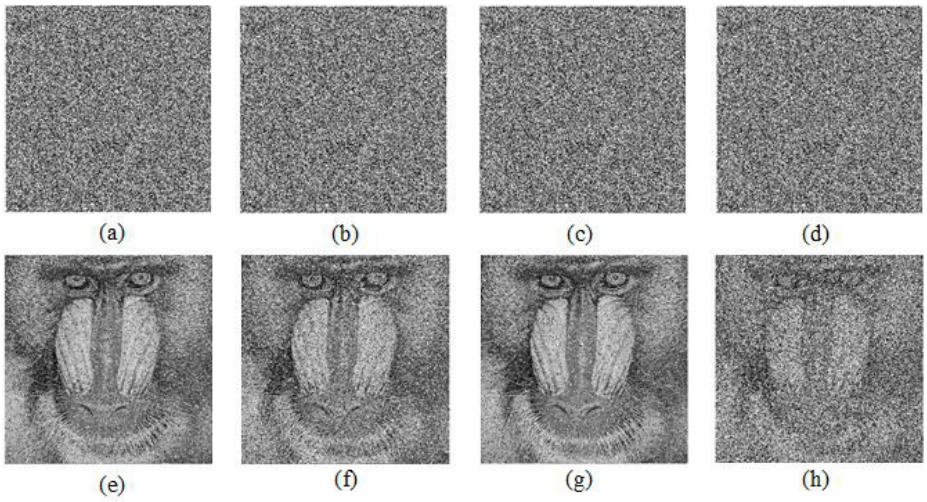

Fig. 18 Noise analysis: (a) corrupted image with SPN with density 0.05; (b) corrupted image with SPN with density 0.1; (c) corrupted image with GN with density 10^{-6}; (d) corrupted image with GN with density 3×10^{-6} (e) decrypted Baboon of (a); (f) decrypted Baboon of (b); (g) decrypted Baboon of(c); (h) decrypted Baboon of (d)

References

1. Wang, M., Wang, X., Zhao, T., Zhang, C., Xia, Z.Yao, N.: Spatiotemporal chaos in improved cross coupled map lattice and its application in a bit-level image encryption scheme, Information Sciences, **544** , 1-24, (2021). https://doi.org/10.1016/j.ins.2020.07.051
2. Xu, C., Sun, J.Wang, C.: An Image Encryption Algorithm Based on Random Walk and Hyperchaotic Systems, International Journal of Bifurcation and Chaos, **30**(04), 2050060, (2020). https://doi.org/10.1142/s0218127420500601
3. Zhou, Y., Li, C., Li, W., Li, H., Feng, W.Qian, K.: Image encryption algorithm with circle index table scrambling and partition diffusion, Nonlinear Dynamics, **103**(2), 2043-2061, (2021). https://doi.org/10.1007/s11071-021-06206-8
4. Zhou, J., Li, J.Di, X.: A Novel Lossless Medical Image Encryption Scheme Based on Game Theory With Optimized ROI Parameters and Hidden ROI Position, IEEE Access, **8**, 122210-122228, (2020). https://doi.org/10.1109/access.2020.3007550
5. Wu, Y., Zhang, L., Qian, T., Liu, X.Xie, Q.: Content-adaptive image encryption with partial unwinding decomposition, Signal Processing, **181**, 107911, (2021). https://doi.org/10.1016/j.sigpro.2020.107911
6. Hua, Z., Zhang, K., Li, Y.Zhou, Y.: Visually secure image encryption using adaptive-thresholding sparsification and parallel compressive sensing, Signal Processing, **183**(4), 107998, (2021). https://doi.org/10.1016/j.sigpro.2021.107998
7. Zhu, H., Dai, L., Liu, Y.Wu, L.: A three-dimensional bit-level image encryption algorithm with Rubik's cube method, Mathematics and Computers in Simulation, **185**, 754-770, (2021). https://doi.org/10.1016/j.matcom.2021.02.009
8. Ye, G., Pan, C., Dong, Y., Shi, Y.Huang, X.: Image encryption and hiding algorithm based on compressive sensing and random numbers in-

sertion, Signal Processing, **172**, 107563, (2020). https://doi.org/10.1016/j.sigpro.2020.107563
9. Naim, M., Ali Pacha, A.Serief, C.: A novel satellite image encryption algorithm based on hyperchaotic systems and Josephus problem, Advances in Space Research, **67**(7), 2077-2103, (2021). https://doi.org/10.1016/j.asr.2021.01.018
10. Coppersmith, D.: The Data Encryption Standard (DES) and its strength against attacks, IBM J. Res. Dev., **38**(3), 243-250, (1994). https://doi.org/10.1147/rd.383.0243
11. Paar, C.Pelzl, J.: The Advanced Encryption Standard (AES), Understanding cryptography, 87-121, (2010).
12. Lai, X.Massey, J. L.: A Proposal for a New Block Encryption Standard, Workshop on the Theory and Application of of Cryptographic Techniques, (1990).
13. Chen, L.-p., Yin, H., Yuan, L.-g., Lopes, A. M., Machado, J. A. T.Wu, R.-c.: A novel color image encryption algorithm based on a fractional-order discrete chaotic neural network and DNA sequence operations, Frontiers of Information Technology & Electronic Engineering, **21**(6), 866-879, (2020). https://doi.org/10.1631/fitee.1900709
14. Sahasrabuddhe, A.Laiphrakpam, D. S.: Multiple images encryption based on 3D scrambling and hyperchaotic system, Information Sciences, **550**, 252-267, (2021). https://doi.org/10.1016/j.ins.2020.10.031
15. Li, C., Lin, D., Lü, J.Hao, F.: Cryptanalyzing an Image Encryption Algorithm Based on Autoblocking and Electrocardiography, IEEE multimedia, (2018). https://doi.org/10.1109/MMUL.2018.2873472
16. Talhaoui, M. Z.Wang, X.: A new fractional one dimensional chaotic map and its application in high-speed image encryption, Information Sciences, **550**, 13-26, (2021). https://doi.org/10.1016/j.ins.2020.10.048
17. Matthews, R.: On the Derivation of a "Chaotic" Encryption Algorithm, Cryptologia, **13**(1), 29-42, (1989). https://doi.org/10.1080/0161-118991863745
18. Yang, F., Mou, J., Ma, C.Cao, Y.: Dynamic analysis of an improper fractional-order laser chaotic system and its image encryption application, Optics and Lasers in Engineering, **129**, 106031, (2020). https://doi.org/10.1016/j.optlaseng.2020.106031
19. Chai, X., Gan, Z., Yuan, K., Chen, Y.Liu, X.: A novel image encryption scheme based on DNA sequence operations and chaotic systems, Neural Computing and Applications, 31(1), 219-237, (2017). https://doi.org/10.1007/s00521-017-2993-9
20. Wang, X., Wang, Y., Zhu, X.Unar, S.: Image encryption scheme based on Chaos and DNA plane operations, Multimedia Tools and Applications, **78**(18), 26111-26128, (2019). https://doi.org/10.1007/s11042-019-07794-9
21. Zhu, Z.-l., Zhang, W., Wong, K.-w.Yu, H.: A chaos-based symmetric image encryption scheme using a bit-level permutation, Information Sciences, **181**(6), 1171-1186, (2011). https://doi.org/10.1016/j.ins.2010.11.009
22. Khan, M., Masood, F., Alghafis, A., Amin, M.Batool Naqvi, S. I.: A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and Brownian motion, PLoS One, **14**(12), e0225031, (2019). https://doi.org/10.1371/journal.pone.0225031
23. Wang, J., Li, J., Di, X., Zhou, J.Man, Z.: Image Encryption Algorithm Based on Bit-Level Permutation and Dynamic Overlap Diffusion, IEEE Access, **8**, 160004-160024, (2020). https://doi.org/10.1109/access.2020.3020187
24. Feng, W., He, Y.-G., Li, H.-M.Li, C.-L.: Image encryption algorithm based on discrete logarithm and memristive chaotic system, The European Physical Journal Special Topics, **228**(10), 1951-1967, (2019). https://doi.org/10.1140/epjst/e2019-800209-3
25. Ye, G.: Image scrambling encryption algorithm of pixel bit based on chaos map, Pattern Recognition Letters, **31**(5), 347-354, (2010). https://doi.org/10.1016/j.patrec.2009.11.008
26. Li, X., Xie, Z., Wu, J.Li, T.: Image Encryption Based on Dynamic Filtering and Bit Cuboid Operations, Complexity, **2019**, 1-16, (2019). https://doi.org/10.1155/2019/7485621
27. Liu, X., Xiao, D.Liu, C.: Quantum image encryption algorithm based on bit-plane permutation and sine logistic map, Quantum Information Processing, **19**(8), (2020). https://doi.org/10.1007/s11128-020-02739-w
28. Laredo, D., Ma, S. F., Leylaz, G., Schütze, O.Sun, J.-Q.: Automatic model selection for fully connected neural networks, International Journal of Dynamics and Control, **8**(4), 1063-1079, (2020). https://doi.org/10.1007/s40435-020-00708-w
29. Mcculloch, W. S.Pitts, W. H.: A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics, **5**, 115-133, (1988).
30. Lu, ZhuWang: A Novel S-Box Design Algorithm Based on a New Compound Chaotic System, Entropy, **21**(10), 1004, (2019). https://doi.org/10.3390/e21101004
31. Phatak, S. C.Rao, S. S.: Logistic map: A possible random-number generator, Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics, **51**(4), 3670-3678, (1995). https://doi.org/10.1103/physreve.51.3670
32. Liu, X., Song, Y.Jiang, G.-P.: Hierarchical Bit-Level Image Encryption Based on Chaotic Map and Feistel Network, International Journal of Bifurcation and Chaos, **29**(02), 1950016, (2019). https://doi.org/10.1142/s0218127419500160
33. Hua, Z., Zhou, Y.Huang, H.: Cosine-transform-based chaotic system for image encryption, Information Sciences, **480**, 403-419, (2019). https://doi.org/10.1016/j.ins.2018.12.048
34. Palacios-Luengas, L., Delgado-Gutiérrez, G., Díaz-Méndez, J. A.Vázquez-Medina, R.: Symmetric cryptosystem based on skew tent map, Multimedia Tools and Applications, **77**(2), 2739-2770, (2017). https://doi.org/10.1007/s11042-017-4375-9
35. Gong, L., Qiu, K., Deng, C.Zhou, N.: An optical image compression and encryption scheme based on compressive sensing and RSA algorithm, Optics and Lasers in Engineering, **121**, 169-180, (2019). https://doi.org/10.1016/j.optlaseng.2019.03.006
36. Nepomuceno, E. G., Nardo, L. G., Arias-Garcia, J., Butusov, D. N.Tutueva, A.: Image encryption based on the pseudo-orbits from 1D chaotic map, Chaos, **29**(6), 061101, (2019). https://doi.org/10.1063/1.5099261
37. Wang, J., Zhi, X., Chai, X.Lu, Y.: Chaos-based image encryption strategy based on random number embedding and DNA-level self-adaptive permutation and diffusion, Multimedia Tools and Applications, (2021). https://doi.org/10.1007/s11042-020-10413-7
38. Zhang, Y., Chen, A., Tang, Y., Dang, J.Wang, G.: Plaintext-related image encryption algorithm based on perceptron-like network, Information Sciences, **526**, 180-202, (2020). https://doi.org/10.1016/j.ins.2020.03.054
39. Xu, Q., Sun, K., He, S.Zhu, C.: An effective image encryption algorithm based on compressive sensing and 2D-SLIM, Optics and Lasers in Engineering, **134**, 106178, (2020). https://doi.org/10.1016/j.optlaseng.2020.106178
40. Xian, Y.Wang, X.: Fractal sorting matrix and its application on chaotic image encryption, Information Sciences, **547**, 1154-1169, (2021). https://doi.org/10.1016/j.ins.2020.09.055

41. Albahrani, E. A., Maryoosh, A. A.Lafta, S. H.: Block image encryption based on modified playfair and chaotic system, Journal of Information Security and Applications, **51**, 102445, (2020). https://doi.org/10.1016/j.jisa.2019.102445
42. Nematzadeh, H., Enayatifar, R., Yadollahi, M., Lee, M.Jeong, G.: Binary search tree image encryption with DNA, Optik, **202**, 163505, (2020). https://doi.org/10.1016/j.ijleo.2019.163505
43. Zhang, Y.: A new unified image encryption algorithm based on a lifting transformation and chaos, Information Sciences, **547**, 307-327, (2021). https://doi.org/10.1016/j.ins.2020.07.058
44. Wang, X.Gao, S.: Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network, Information Sciences, **539**, 195-214, (2020). https://doi.org/10.1016/j.ins.2020.06.030
45. Wu, Y., Noonan, J. P.Agaian, S.: NPCR and UACI randomness tests for image encryption, Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT), **1**(2), 31-38, (2011).

Figures

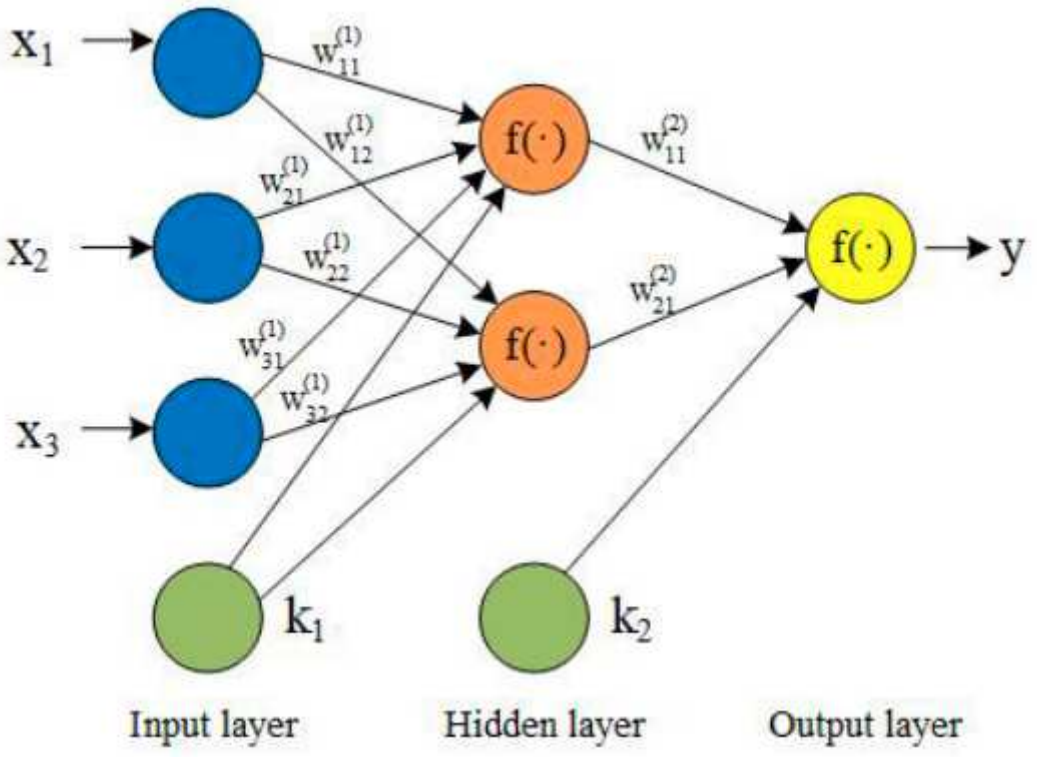

Figure 1

Please see the Manuscript PDF file for the complete figure caption

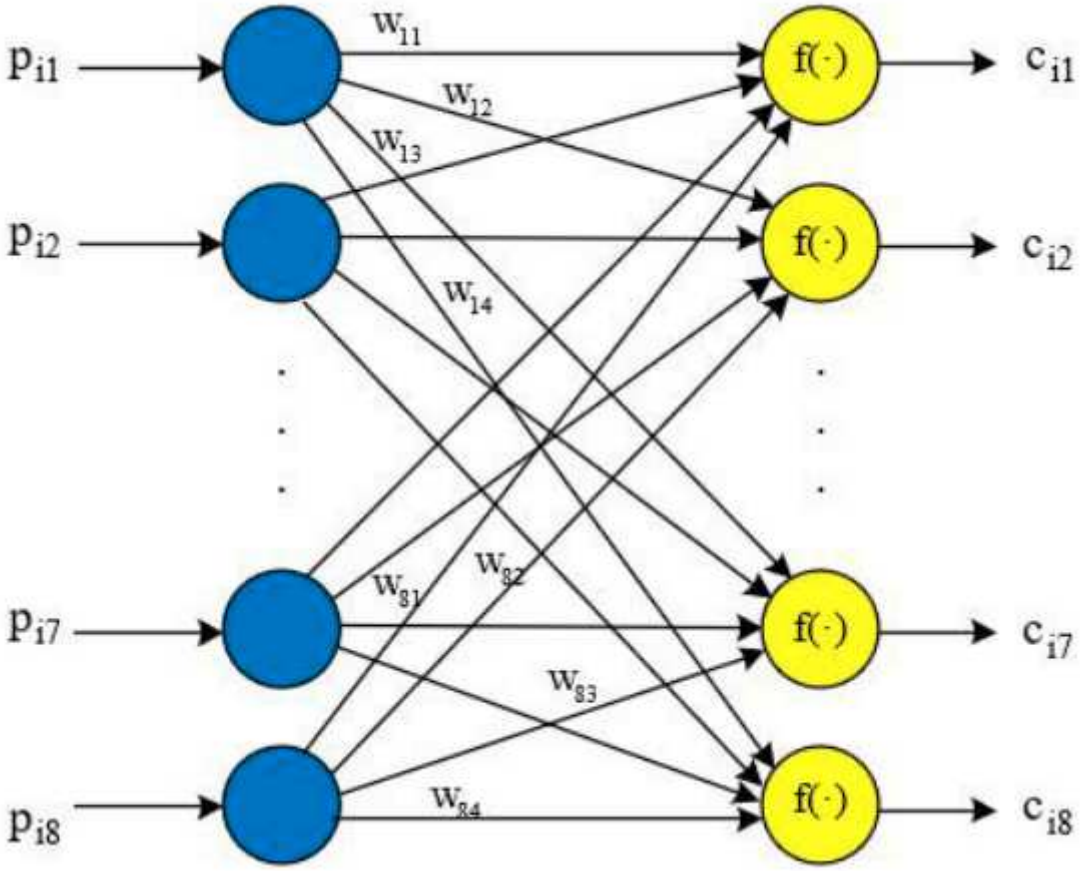

Figure 2

Please see the Manuscript PDF file for the complete figure caption

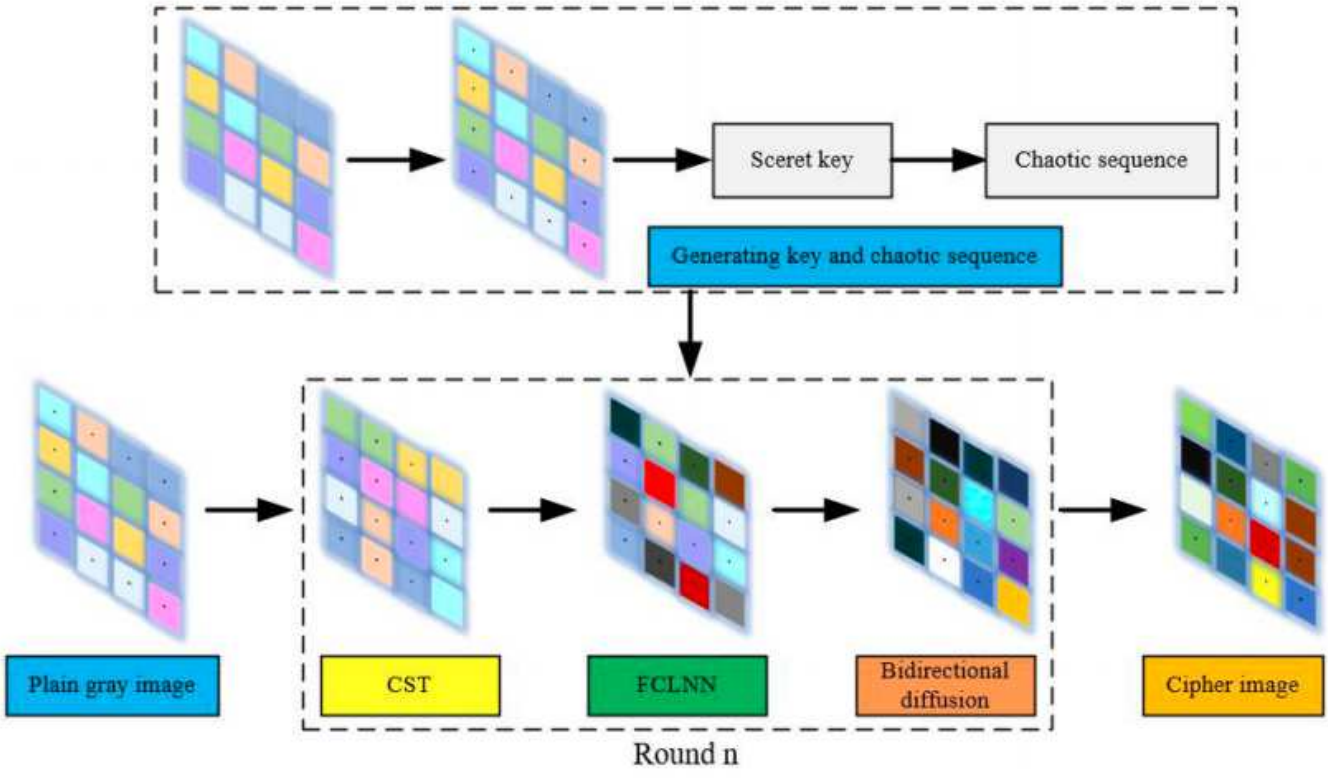

Figure 3

Please see the Manuscript PDF file for the complete figure caption

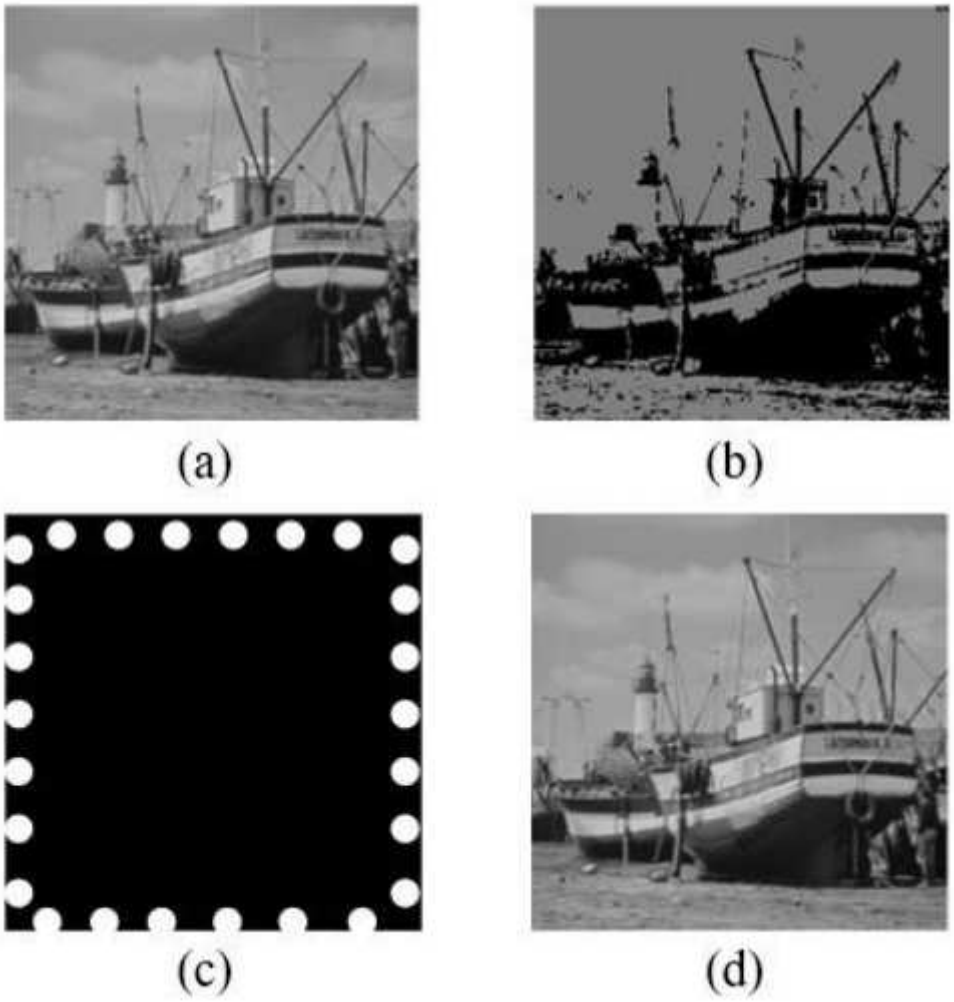

Figure 4

Please see the Manuscript PDF file for the complete figure caption

138	138	...	127	126
138				127
...				...
145				128
125	125	...	141	136

First time

139	139	...	127	126
138				127
...				...
145				128
125	124	...	140	136

Second time

138	138	...	126	126
138				126
...				...
145				129
125	124	...	141	136

Third time

Figure 5

Please see the Manuscript PDF file for the complete figure caption

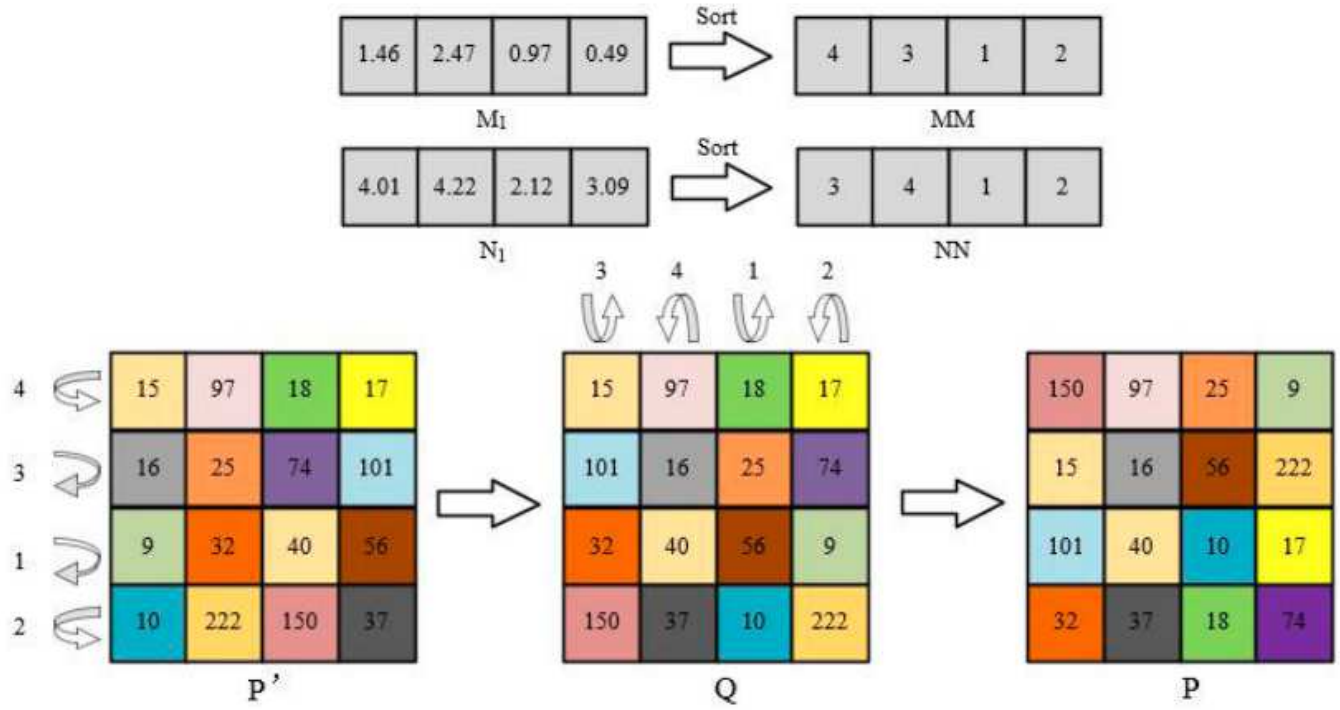

Figure 6

Please see the Manuscript PDF file for the complete figure caption

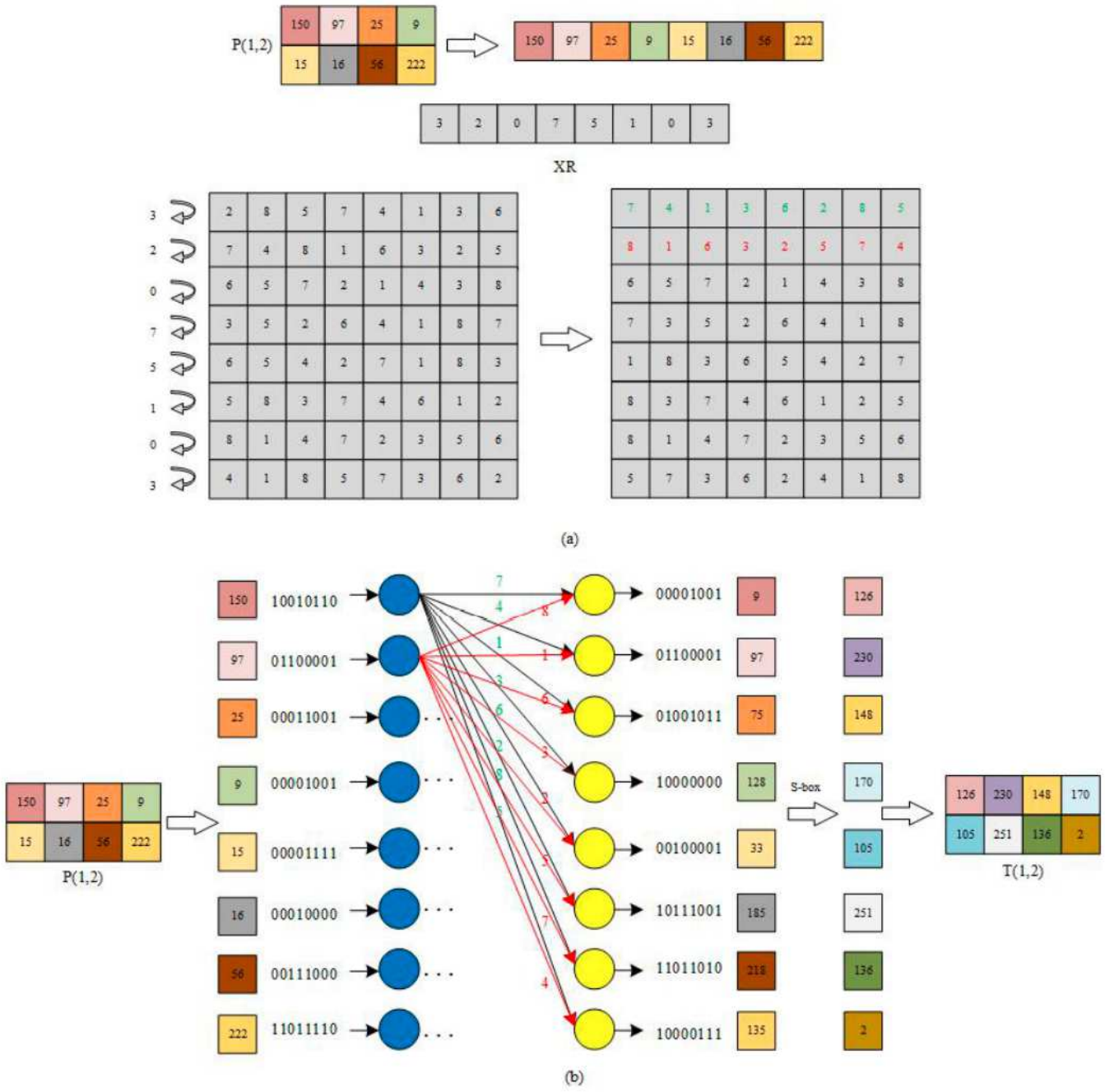

Figure 7

Please see the Manuscript PDF file for the complete figure caption

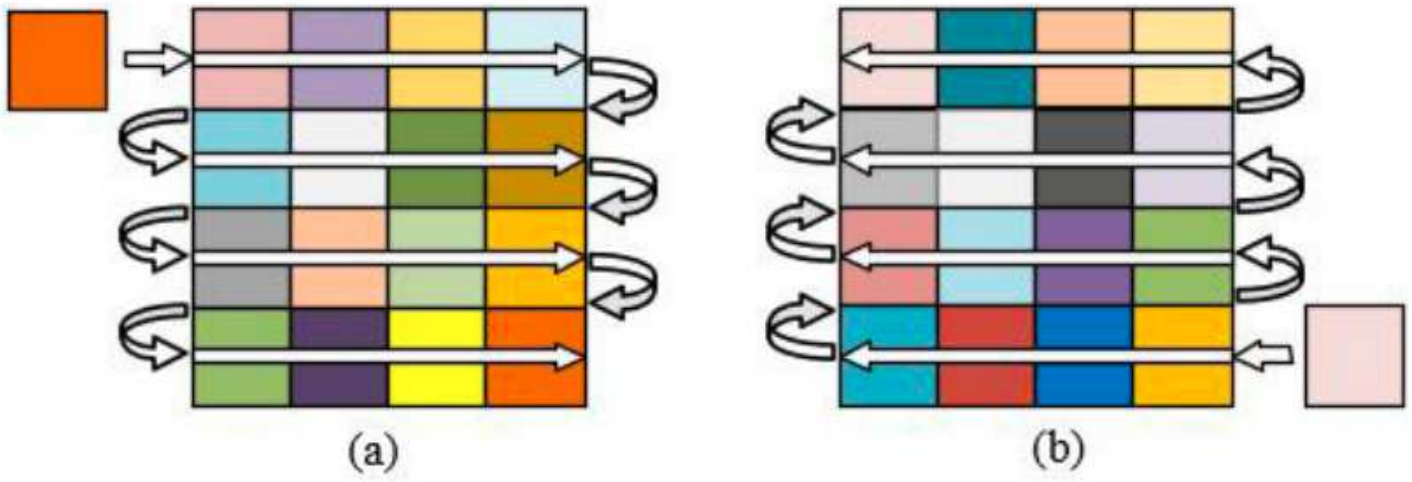

Figure 8

Please see the Manuscript PDF file for the complete figure caption

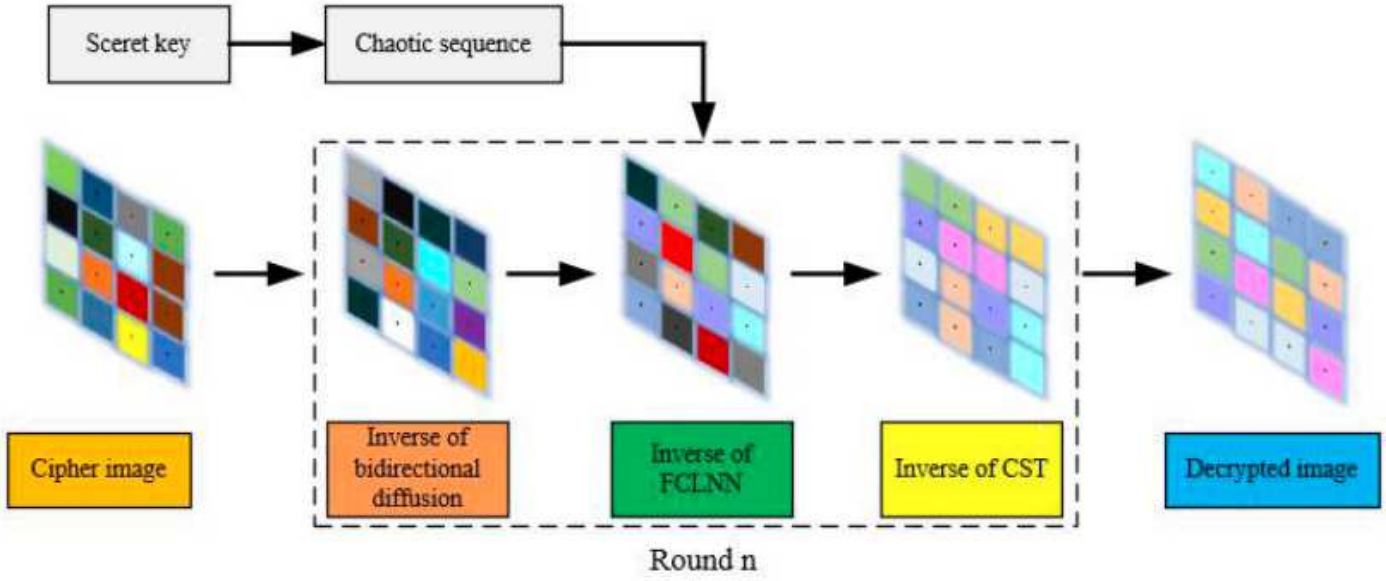

Figure 9

Please see the Manuscript PDF file for the complete figure caption

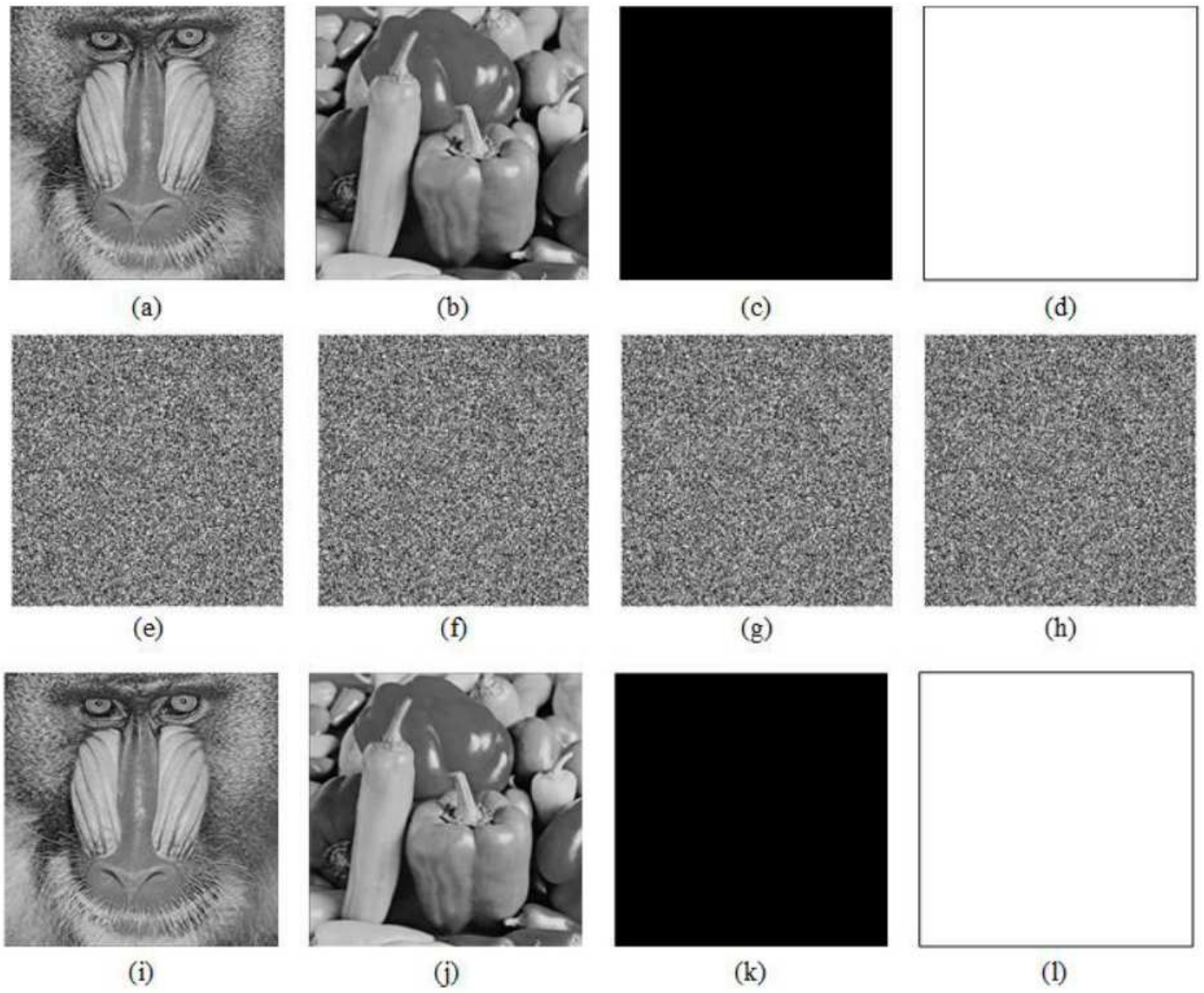

Figure 10

Please see the Manuscript PDF file for the complete figure caption

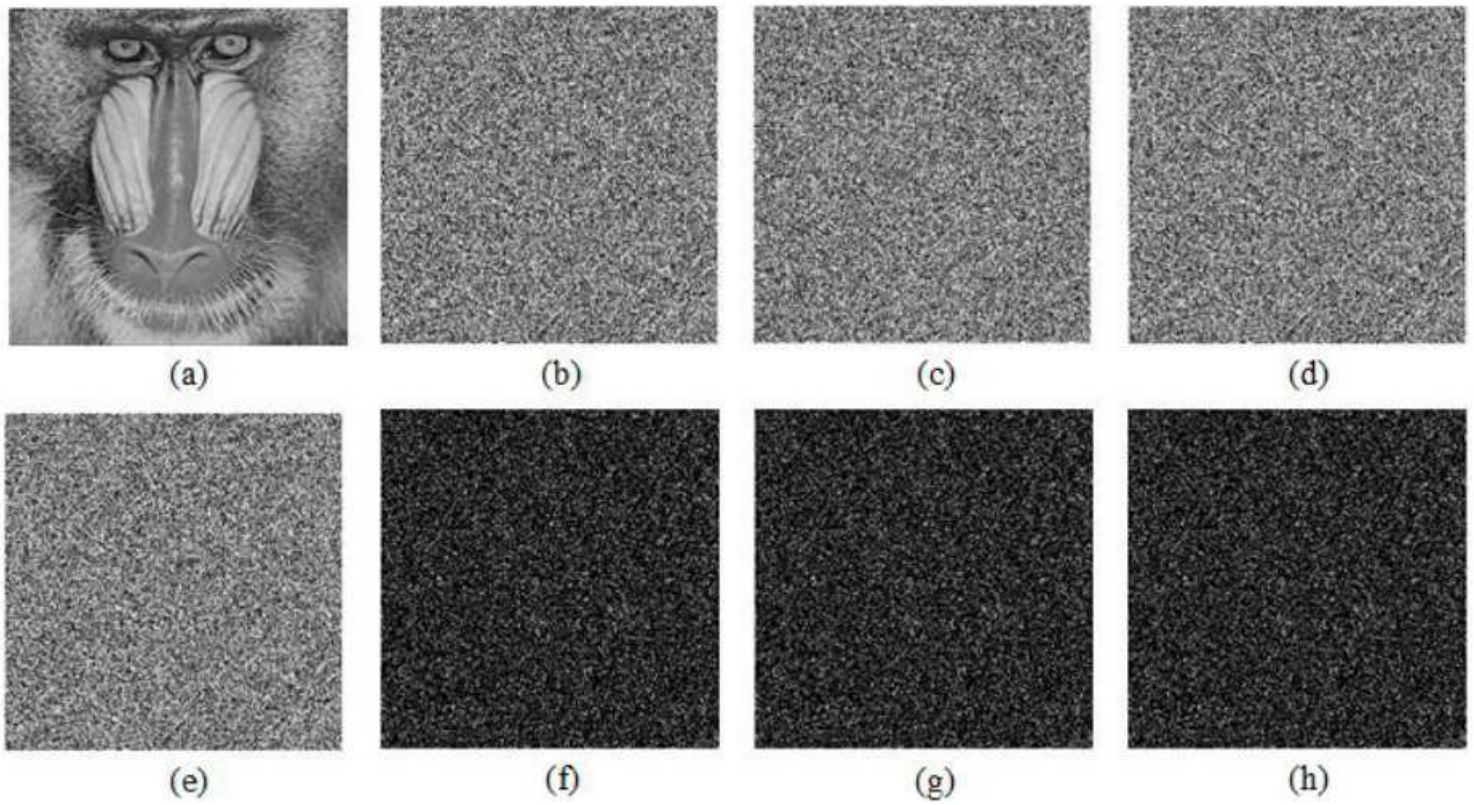

Figure 11

Please see the Manuscript PDF file for the complete figure caption

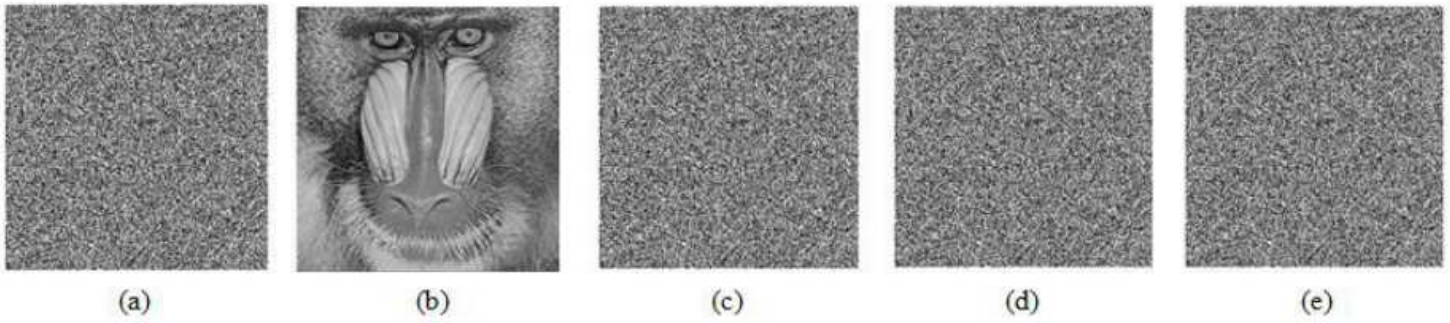

Figure 12

Please see the Manuscript PDF file for the complete figure caption

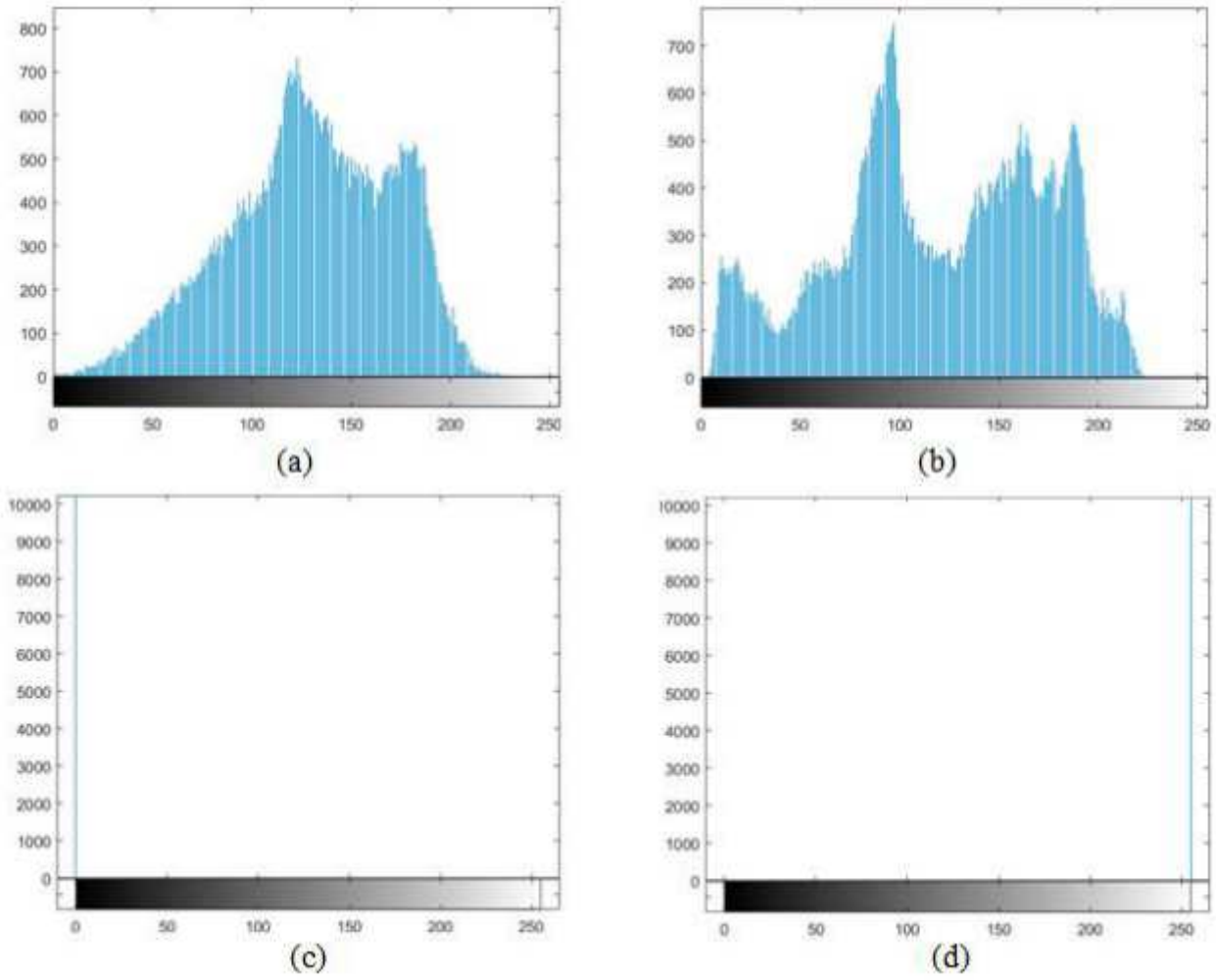

Figure 13

Please see the Manuscript PDF file for the complete figure caption

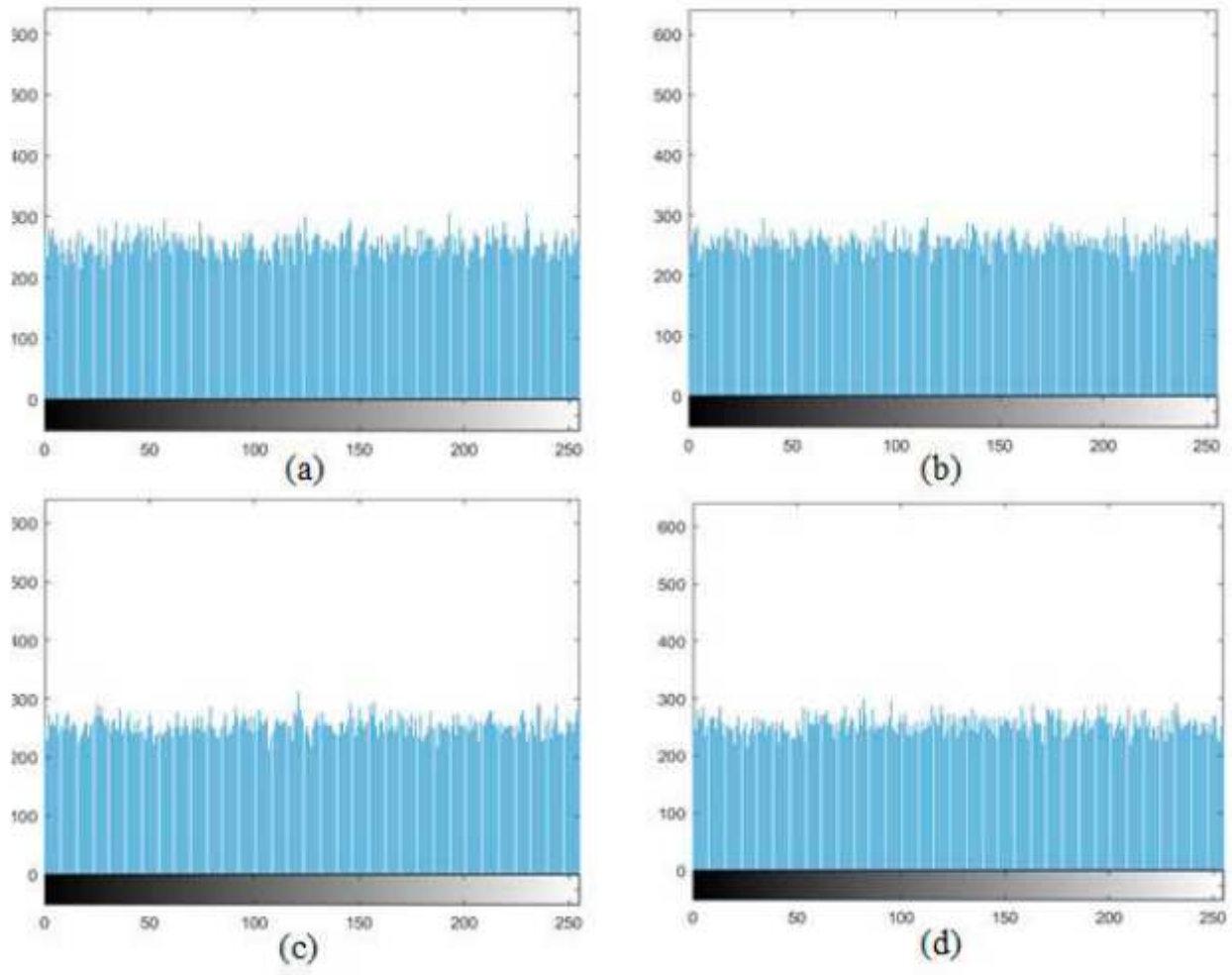

Figure 14

Please see the Manuscript PDF file for the complete figure caption

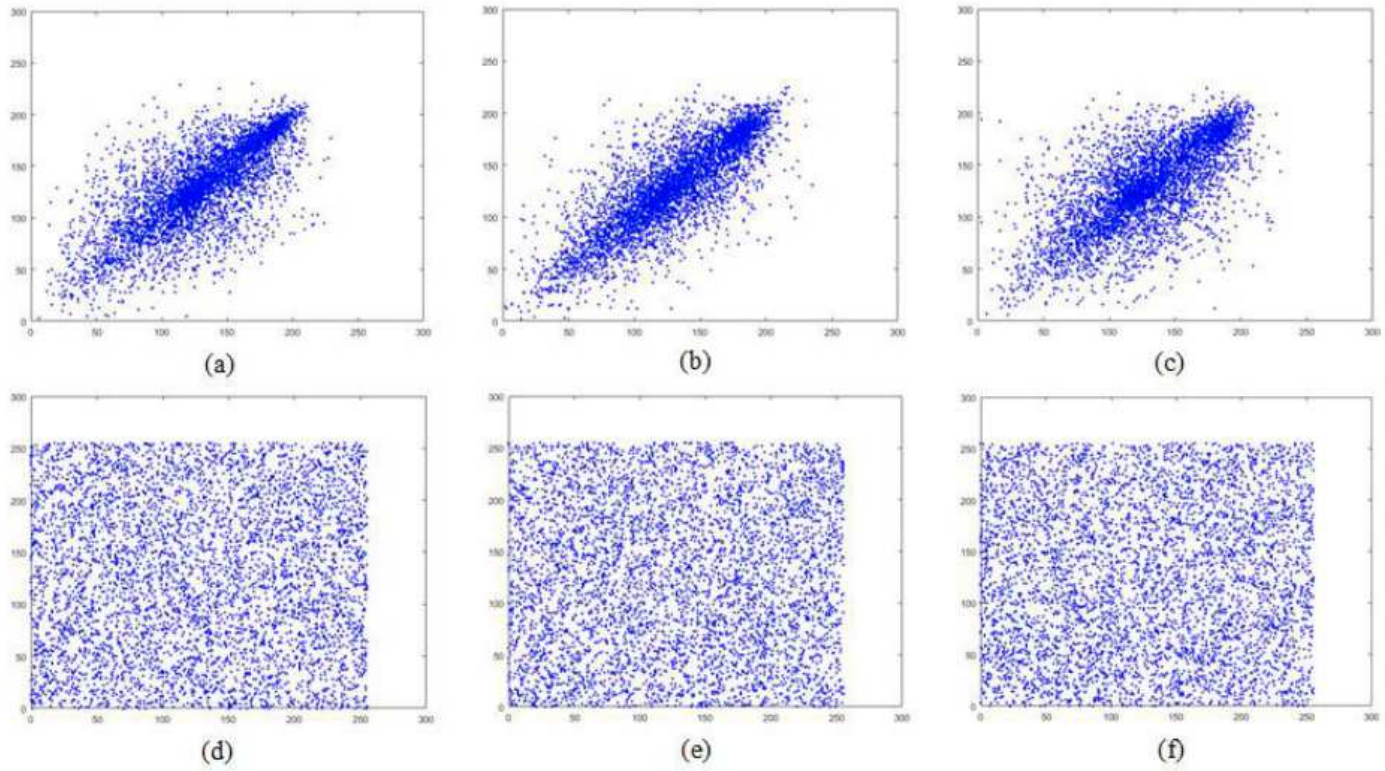

Figure 15

Please see the Manuscript PDF file for the complete figure caption

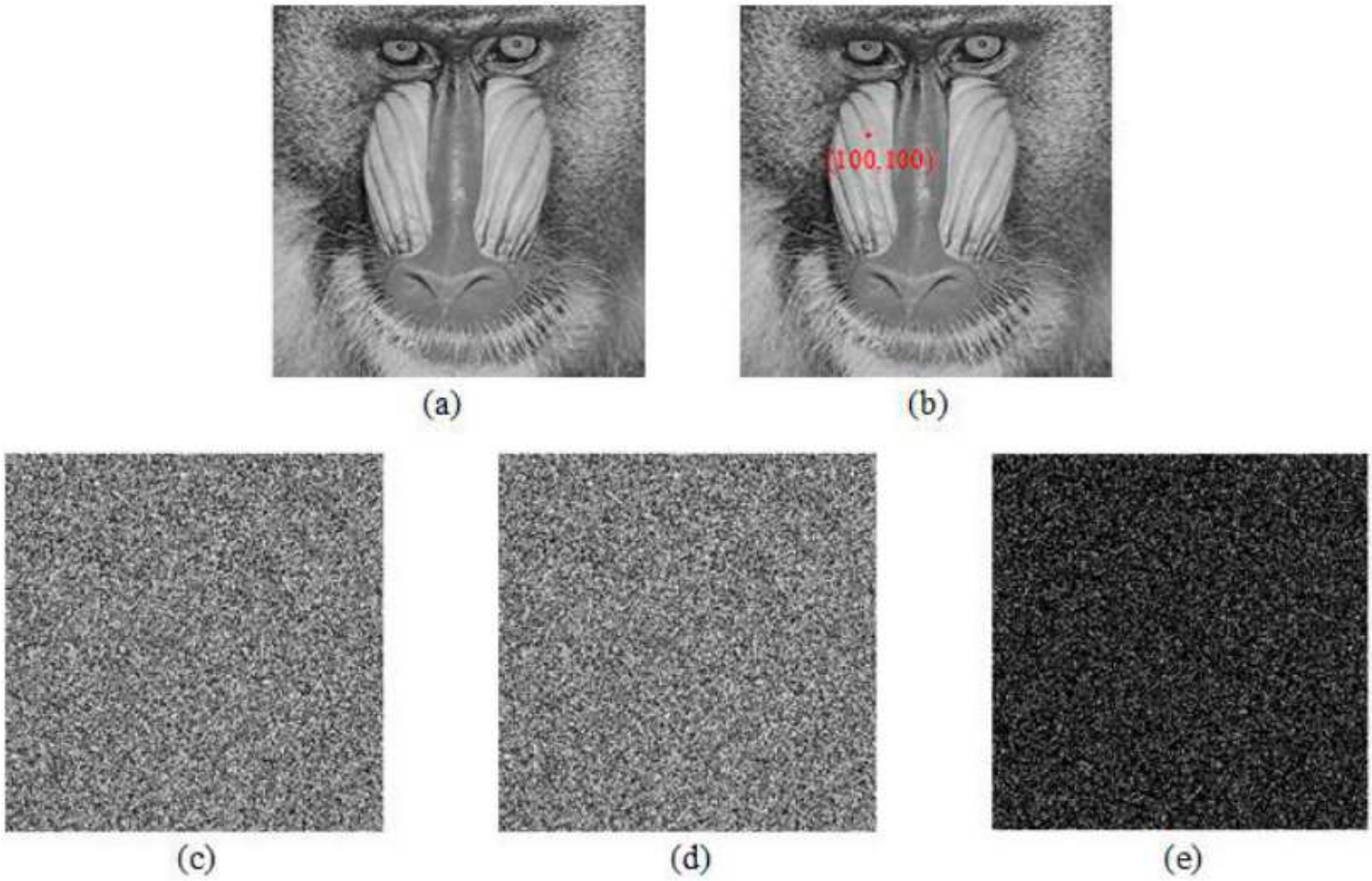

Figure 16

Please see the Manuscript PDF file for the complete figure caption

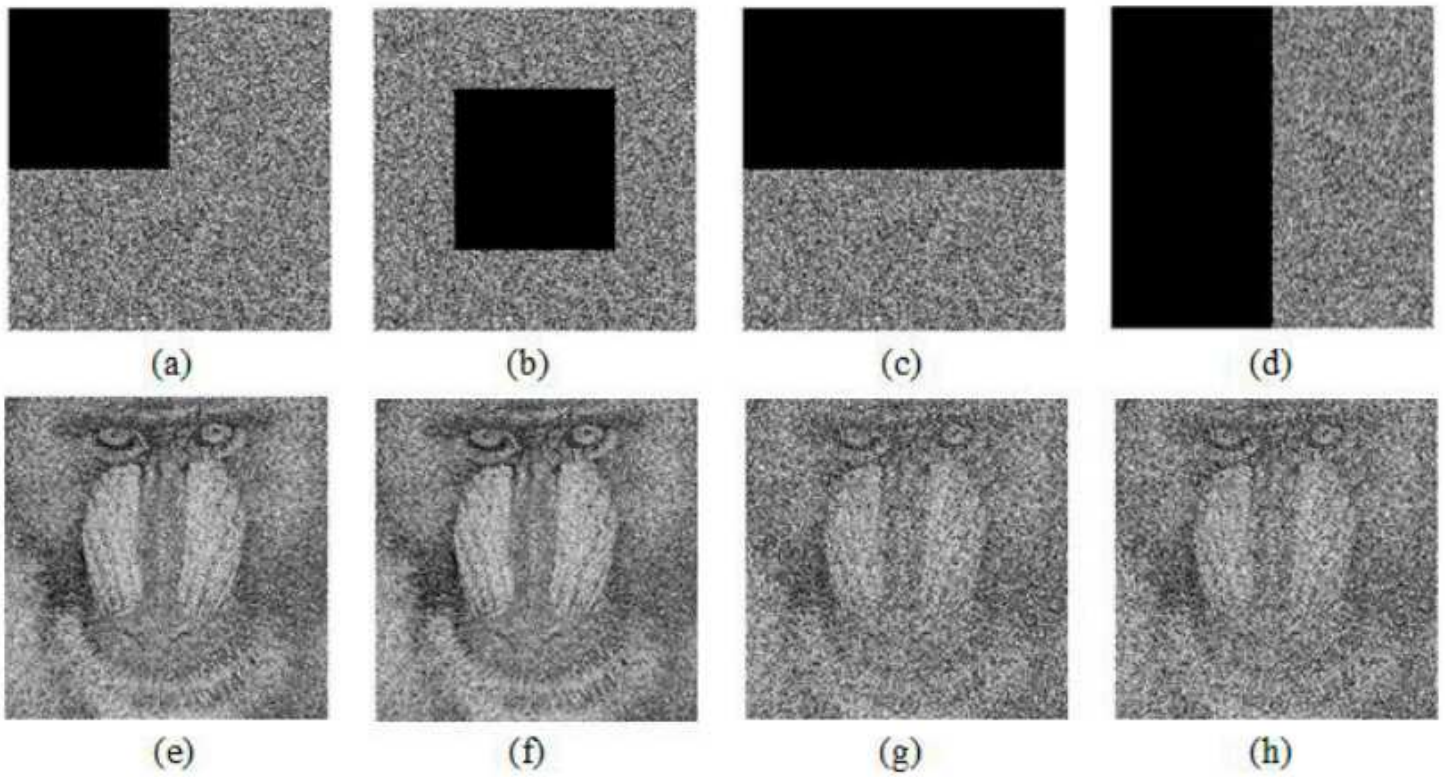

Figure 17

Please see the Manuscript PDF file for the complete figure caption

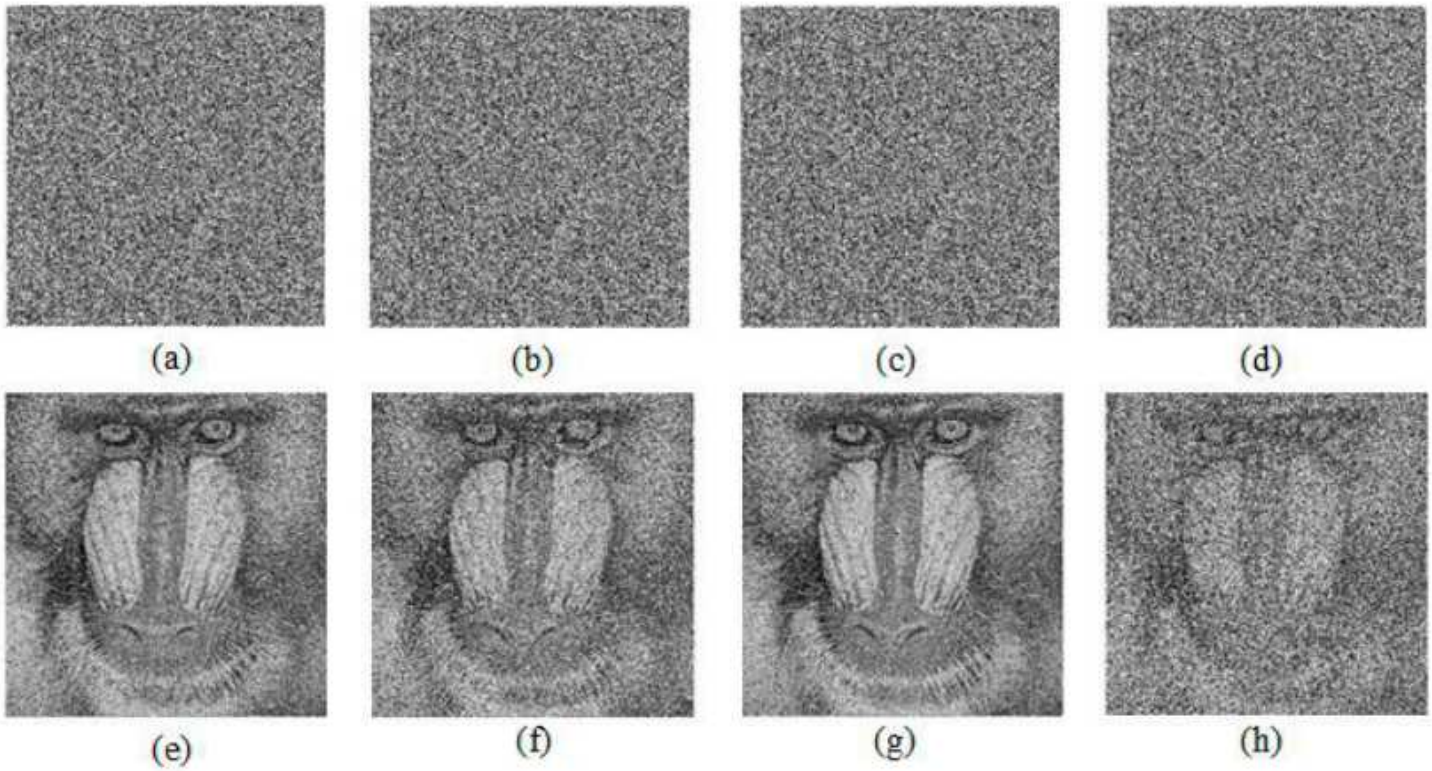

Figure 18

Please see the Manuscript PDF file for the complete figure caption