Hindawi
Security and Communication Networks
Volume 2022, Article ID 6047349, 14 pages
https://doi.org/10.1155/2022/6047349

Research Article

Color Image Encryption Based on Deep Learning and Block Embedding

Yi Liu, Gang Cen, Bijun Xu, and Xiaogang Wang

Zhejiang University of Science and Technology, Hangzhou 310023, China

Correspondence should be addressed to Gang Cen; gcen@zust.edu.cn

Received 26 July 2022; Revised 29 September 2022; Accepted 13 October 2022; Published 26 October 2022

Academic Editor: Je Sen Teh

Because of the advent of the information age, digital multimedia data are mainly transmitted through the Internet. Image is one of the most popular digital multimedia data. Therefore, this paper proposes a block-based key embedding method and a color image encryption scheme based on deep learning with this method. By using a neural network model to predict the initial chaotic sequence, the key data generated by prediction are encrypted to the color image in layers and blocks. This paper proposed the color image encryption scheme time complexity $O(n^2)$, which is simple and effective. Simulation results show that the proposed scheme exhibits good statistical properties with information entropy mean close to eight and correlation coefficient close to zero. Good robustness to common image attacks like noise addition and cropping. Excellent encryption performance includes enormous key space and low PSNR.

1. Introduction

With the birth and development of fifth-generation mobile communication technology, people can obtain all kinds of data they need through the Internet anytime and anywhere and realize resource sharing and information interaction. Therefore, information security has consistently commanded much attention. Image encryption represents the process of hiding images with keys to prevent unauthorized access. Considering the characteristics of image data, many existing image encryption algorithms are proposed based on different technologies, including vector quantization, fractional wavelet transformation, and chaos [1].

In recent years, deep learning models have been widely used in the field of information security. Especially in image processing, deep learning shows vast advantages (see Figure 1). For instance, Li et al. used convolution neural networks (CNN) to optically encrypt iris images [2]. Chen et al. proposed a method to improve the robustness of 2D/3D optical image encryption by using extended deep CNN [3]. Ni et al. tried to apply the compressed sensing (CS) reconstruction algorithm based on deep learning to image encryption [4]. Zhang and Li encrypted images using optics combined with deep learning models [5] and so on.

Combining the contents of the above (see Table 1), it is unproblematic to find most of the current encryption schemes based on deep learning which still stay at the stage of encrypting gray image. Based on the research done by Zhao et al. [6], this paper proposes a method of embedding keys in blocks and a color image encryption scheme, depending on a deep learning model to expand the research content in this direction. The deep learning model is principally used to predict the initial sequences generated by chaotic systems by embedding the predicted data into the upper left, lower right, and middle regions of the three components of the color image. The final encrypted image is obtaines by using scrambling and diffusion algorithms.

The main contributions of this paper are as follows:

(1) This is a recent attempt to combine deep learning models with color image encryption

(2) The implementation is simple and effective and improves the practical use efficiency of encryption algorithms

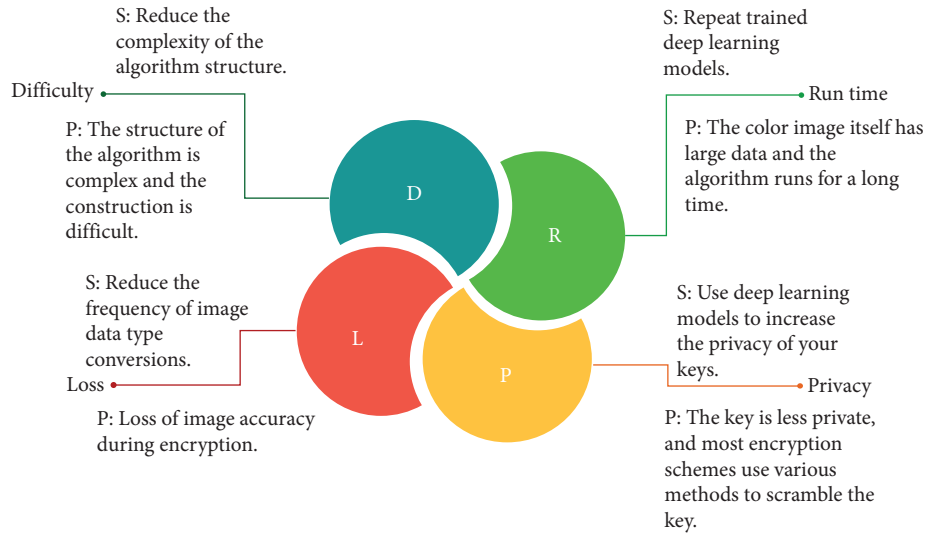

Figure 1: Color image encryption problems and possible solutions for real-world situations.

Table 1: Some image encryption methods based on deep learning.

Method	Proposed	Ref. [2]	Ref. [3]	Ref. [4]	Ref. [6]	Ref. [5]
Image type	Color	Gray	Gray	Gray	Gray	Gray
Model	BiLSTM	CNN	CNN	CNN	LSTM	U-net

(3) It has excellent encryption performance, good statistical characteristics, and strong robustness to common image attacks

2. Preparatory Knowledge

2.1. Chen's Chaotic System. A chaotic system means there are seemingly erratic irregular movements in a certain way which are nonrepeatable, unpredictable, and uncertain. Common three-dimensional chaotic systems include Lorenz and Chen.

In 1999, Professor Chen and Ueta of Houston University put forward the chaotic system [7]. Its equations are defined is given in formula (1).

When the parameters are given $a = 35, b = 3, c \in [20, 28.4]$, Chen's chaotic model presents a chaotic state. It is similar to the Lorenz chaotic model but has a more complex topological structure than the Lorenz chaotic model. Therefore, Chen's chaotic model is widely used in the encryption field. This paper chooses Chen's chaos to generate the initial key sequence:

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = (c - a)x - xz + cy, \\ \dot{z} = xy - bz. \end{cases} \tag{1}$$

2.2. BiLSTM Neural Network. The encryption scheme proposed in this paper can predict the initial sequence generated by a chaotic system based on an arbitrary recurrent neural network. At this place, we choose bidirectional long-short term memory (BiLSTM). BiLSTM is a deep learning model developed based on long-short term memory (LSTM), which realizes more training by traversing the input data twice [8]. It is principally composed of forwarding and backward LSTM. Each time point contains an LSTM unit for selective memory, and forgetting and output information. The LSTM cell formula can be represented as follows:

$$\begin{aligned} I_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ F_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \\ O_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ H_t &= O_t * \tan h(C_t), \\ C_t &= F_t * C_{t-1} + I_t * \widetilde{C}_t, \\ \widetilde{C}_t &= \tan h(W_c \cdot [h_{t-1}, x_t] + b_c). \end{aligned} \tag{2}$$

BiLSTM splices the output of forwarding LSTM and backward LSTM at times $t-1$, t, $t+1$, etc. Because it can use past time and subsequent time to forecast at the same time, it can provide a better prediction effect than LSTM.

2.3. Arnold Transformation. Arnold transformation, also known as cat mapping, was first introduced by Russian mathematician Vladimir Arnold in the study of ergodic theory. This transformation involves recutting and splicing the matrix of digital images [9].

The two-dimensional Arnold transformation of a digital image with $M = N$ (equal length and width) is defined as

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod(N). \tag{3}$$

In formula (3), x_n, y_n represents the position of pixels in the digital image before the transformation. x_{n+1}, y_{n+1} indicates the position of the transformed pixels. a, b are parameters. n represents the number of current

transformations. N is the length or width of the image. Mod$(-)$ represents modular operations.

2.4. The Proposed Block Embedding Algorithm

2.4.1. Theoretical Derivation. A good image encryption algorithm should have statistical characteristics like low correlation and high entropy. Hosny et al. obtained encrypted images with negative correlation and high entropy by chunking and scrambling the three-color subimages of color images and mapping them with one-dimensional chaotic logic maps [10]. Wang et al. employed the improved zigzag method to scramble the subimages after the color image was chunked, and the combination with the chaotic system made the encrypted image more difficult to crack [11]. Younes and Aman experimentally demonstrated that this method can effectively reduce the correlation between the pixels of the encrypted image by dividing the original image into an arbitrary number of blocks [12].

After summarizing the above research results, we can infer that according to the characteristics of the three-color channel of the color image, the channel split of the color image and the chaos of the split subgraph can theoretically and effectively reduce the correlation of adjacent pixels of the image and weaken the structure of the initial image, thereby improving the security of the encryption scheme. As a result, we designed a key embedding method, which mainly embeds the generated key matrix into various regions of different color channels in blocks. The specific embedding idea is shown in Figure 2.

To simplify the explanation, we used the $7 * 7$ matrix as an example in Figure 1. After splitting the color image with uniform length and width into R, G, and B color matrices, the key sequences $x\mathrm{Pr}ed$, $y\mathrm{Pr}ed$, $z\mathrm{Pr}ed$ with a length is $(M = N)M - \mathrm{mod}(M, 10)$ predicted by the deep learning model. The key sequences and the R matrix, G matrix, and B matrix of $M * N$ start from the elements in the upper left corner, lower right corner, and intermediate position, respectively and carry out XOR operation one by one according to the direction sequences marked in the figure. At the last moment, the matrix data are obtained after embedding the key (Algorithm 1).

2.4.2. Pseudocode for the Proposed Block Embedding Algorithm

2.5. The proposed encryption schemes

2.5.1. Theory and Steps. Deep learning refers to artificial neural networks (ANNs) with complex multilayers [13]. LSTM is a specific recurrent neural network (RNN) that solves the long-term dependency problem on RNN [14] so that it can achieve good predictive performance at large time intervals. There have been studies to apply LSTM to the prediction of chaotic time series. Sangiorgio and Dercole have experimentally checked that LSTM networks are superior to feed-forward competitors in predicting chaotic time series and have good robustness [15]. BiLSTM is formed by a combination of LSTMs in the forward and backward directions, which can be seen as a two-layer neural network.

From the above research results, it can be perceived that is theoretically effective to generate robust chaotic sequences using BiLSTM. In the actual application scenario, even if the chaotic system that generates the initial key is cracked by the attacker. The attacker cannot get the final encryption key by reverse inference of the deep learning model. It improves the security of the encryption scheme to a certain extent and can resist some common image attacks.

In the process of using deep learning for color image encryption research, we found there is a certain contradiction between the two aspects of reducing the difficulty of use in actual scenes while being both secure and at the same time. To this end, we have introduced our design of the block embedding method to reduce complex calculations while ensuring the safety and robustness of the algorithm.

The encryption scheme proposed in this paper represents a process of splitting the color image into R, G, and B color matrices, predicting and generating three new chaotic key sequences by BiLSTM, embedding the key sequences into the three-color matrices in blocks, and generating the final encrypted image by scrambling and diffusion. The main encryption process (see Figure 3) and the specific encryption steps are as follows (Algorithm 2):

(1) CauseThe original image F and its size be $M * N(M = N)$. Given the initial value x_0, y_0, z_0 of the Chen chaotic system, the initial chaotic sequences x, y, z are obtained.

(2) The initial chaotic sequences X, Y, Z with length s are intercepted as the dataset of BiLSTM, and the parameters of the deep learning model are set to generate the predicted sequences $x\mathrm{Pr}ed$, $y\mathrm{Pr}ed$, $z\mathrm{Pr}ed$.

(3) The image F is divided into three-color matrices R, G, and B. The sequences $x\mathrm{Pr}ed$, $y\mathrm{Pr}ed$, $z\mathrm{Pr}ed$ are embedded into the three-color matrices by using the block embedding algorithm (see the section Block Embedding). The embedded matrix E_{Im} is obtained.

(4) Given two pseudo-random integers n_1, n_2 with the range of values $[0, 255]$, the E_{Im} is transformed by n_1 Arnold transformation to obtain the color matrices SR, SG, SB after the first scrambling.

(5) After splicing the initial chaotic sequences x, y, z and the predicted sequences $x\mathrm{Pr}ed, y\mathrm{Pr}ed, z\mathrm{Pr}ed$, the new sequences x', y', z' are gained awarding to

$$x', y', z' = \mathrm{mod}\Big(\mathrm{floor}\,(x + x\mathrm{Pr}ed, y + y\mathrm{Pr}ed, z + z\mathrm{Pr}ed) * 2^{16}, 256\Big). \tag{4}$$

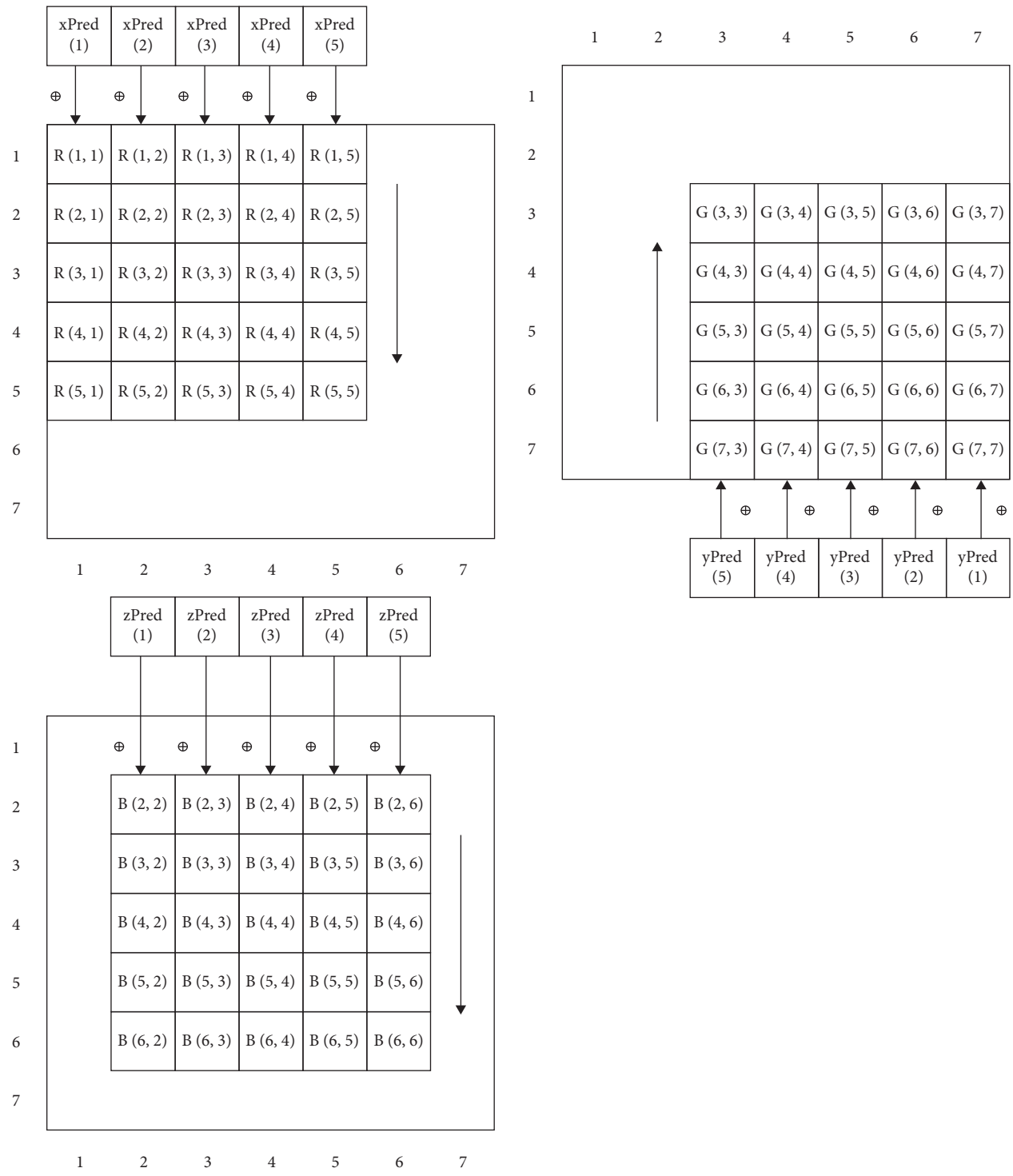

Figure 2: Schematic diagram of three-color channel block embedding key.

Input: R, G, B channels of image F and keys $x\text{Pre } d, y\text{Pre } d, z\text{Pre } d$
Output: matrix e_Im.
Step 1: the R channel is embedded using $x\text{Pre } d$.
  For $i = 1$: len
    For $j = 1$: len
      Temp $= R_{i,j}$
      $R_{i,j} = \text{Temp} \oplus x\text{Pre}d_j$
    End
  End
Step 2: the G channel is embedded using $y\text{Pre } d$.
  For $i = 1$: len
    For $j = 1$: len
      Temp $= G_{G\_\text{Size}_1 - i+1, G\_\text{Size}_1 - j+1}$
      $G_{G\_\text{Size}_1 - i+1, G\_\text{Size}_1 - j+1} = \text{Temp} \oplus y\text{Pre}d_j$
    End
  End
Step 3: the B channel is embedded using $z\text{Pre } d$.
  $M = \text{mod}(B\_\text{Size}_1, 10)$
  $M_{\text{coord}} = M/2$
  For $i = 1$: len
    For $j = 1$: len
      Temp $= B_{i+M_{\text{coord}}, j+M_{\text{coord}}}$
      $B_{i+M_{\text{coord}}, j+M_{\text{coord}}} = \text{Temp} \oplus z\text{Pre}d_j$
    End
  End
Step 4: the matrix e_Im is obtained by concatenating the three-color channels
Step 5: encapsulate it as an embedding method and name it embedKey

Algorithm 1: Pseudocode of the proposed block embedding algorithm.

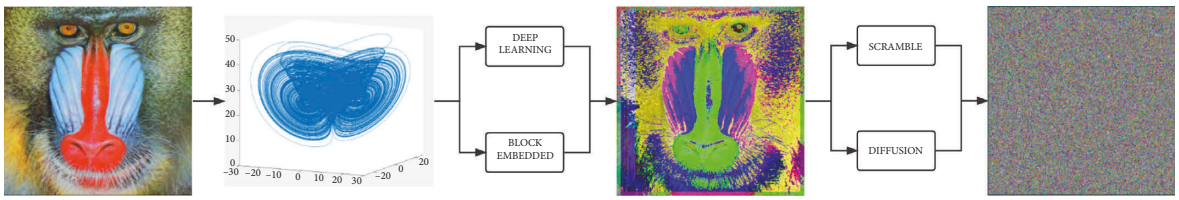

Figure 3: Flow chart of the encryption scheme.

The sequences x', y', z' are used for forwarding diffusion and reverse diffusion of the three-color matrices SR, SG, SB according to formula (5). On that occasion, the diffused matrices R', G', B' are obtained as

$$R' = \begin{cases} RF_i = RF_{i-1} \oplus x'_i \oplus SR_i \\ RB_i = RB_{i+1} \oplus y'_i \oplus RF_i \end{cases}, G' = \begin{cases} GF_i = GF_{i-1} \oplus x'_i \oplus SG_i \\ GB_i = GB_{i+1} \oplus z'_i \oplus GF_i \end{cases}, B' = \begin{cases} BF_i = BF_{i-1} \oplus y'_i \oplus SB_i \\ BB_i = BB_{i+1} \oplus z'_i \oplus BF_i \end{cases}. \quad (5)$$

(6) n_2 Arnold transformations are carried out on the matrices R', G', B', and the final encrypted image E is obtained by splicing the transformed three matrices.

2.5.2. Pseudocode for the Proposed Image Encryption Algorithm

2.6. Decryption Scheme. The decryption scheme in this paper represents the reverse process of an encryption scheme. The detailed steps of the decryption scheme remain as follows (Algorithm 3):

(1) Performing n_2 inverse Arnold transforms an encrypted image E to obtain a diffusion image

(2) After antidiffusion of the diffusion image, the n_1 inverse Arnold transform is carried out to obtain the embedded image reE_{Im}

(3) After the inverse operation of the embedded image, the restored image reF is obtained

Input: original image F
Output: encrypted image E and keys $x\text{Pre } d, y\text{Pre } d, z\text{Pre } d, x', y', z'$
Step 1: $x, y, z = \text{Chen}(x_0, y_0, z_0)$
Step 2: $X, Y, Z = x, y, z(1{:}\ s)$
$x\text{Pre } d, y\text{Pre } d, z\text{Pre } d = \text{BiLSTM}(X, Y, Z)$
Step 3: get E_{Im} using the custom embedding method embedKey
$E_{Im} = \text{embedKey}(F, x\text{Pre } d, y\text{Pre } d, z\text{Pre } d)$
Step 4: $\text{SR}, \text{SG}, \text{SB} = \text{ART}(E_{IM}(1, 2, 3), n_1)$
Step 5: the matrix R', G', B' after diffusion is obtained by using x', y', z' to diffuse SR, SG, SB in the forward and backward direction as
$x', y', z' = \text{mod}(\text{floor}(x + x\text{Pre } d, y + y\text{Pre } d, z + z\text{Pre } d) * 2^{16}, 256)$
$\text{RF}_1 = (0 \oplus x_1') \oplus \text{SR}_1$
For $i = 2{:}\ M * N$
$\text{RF}_i = (\text{RF}_{i-1} \oplus x_i') \oplus \text{SR}_i$
End
$\text{RB}_{M*N} = (0 \oplus y_{M*N}') \oplus \text{RF}_{M*N}$
For $i = M * N - 1{:}\ -1{:}\ 1$
$\text{RB}_i = (\text{RB}_{i+1} \oplus y_i') \oplus \text{RF}_i$
End
$R' = \text{reshape}(\text{RB}, M, N)$
. . . . . .
Step 6: $E = \text{ART}(R', G', B', n_2)$

Algorithm 2: Pseudocode of the proposed image encryption algorithm.

2.6.1. Pseudocode for the Image Decryption Algorithm.

3. Simulation Experiment Results and Analysis

A good encryption scheme should consider good statistical characteristics of digital images, good robustness against common image attacks, and an enormous key space. In this paper, MATLAB is used to simulate the proposed encryption scheme. Through the analysis and discussion of the experimental results, it has been proved that the scheme can effectively encrypt and decrypt color images.

The main pictures used in this section come from the open-source website (https://sipi.usc.edu/database/). The image sizes are $256 * 256$, $512 * 512$, and $1024 * 1024$, respectively. The encryption and decryption effects of the three groups of experimental color images (see Figure 4).

3.1. Statistical Characteristic Analysis

3.1.1. Histogram. An image histogram obtains a graphical expression to reflect the intensity distribution of pixels in the image. Putting differently, it is to count the number of each pixel block in the image, which reflects the most essential statistical characteristics of the image [16]. The flatter the histogram drawn by encrypting the image, the more uniform the pixel value distribution in the image, the smaller the analysis space left for attackers and the better the encryption performance of the image.

In this paper, three groups of experimental objects are encrypted, and the encrypted images are tested by histogram. The results are shown in Figure 5. From the histogram test results, we can observe the encryption scheme proposed in this paper, make the pixel distribution in the encrypted image become uniform, and effectively reduce the analysis space of attackers.

3.1.2. Adjacent Pixel Correlation. Adjacent pixel correlation refers to the relationship diagram between adjacent pixels drawn by randomly selecting N pixels in an image and using the pixel values of two adjacent pixels as horizontal and vertical coordinates, respectively. The more the points in the graph are that remain concentrated near the diagonal of the coordinate axis, the stronger the correlation between the adjacent pixels of the image is. On the contrary, the more the points in the graph are dispersed in the whole graph, the weaker the correlation between adjacent pixels of the image. A good encryption scheme should be able to effectively reduce the correlation between the adjacent pixels of the image [17]. In the field of image encryption, the correlation analysis of adjacent pixels of encryption schemes is usually carried out from three directions horizontal, vertical, and diagonal.

The correlation analysis of adjacent pixels of this encryption scheme is shown in Figure 6. Observing the analysis chart, we can see that the correlation between adjacent pixels of the image encrypted by the encryption scheme proposed in this paper is surely reduced after measuring the three directions, i.e., horizontal, vertical and diagonal. We can find that the drawing points are evenly distributed in the whole analysis map, which can effectively resist statistical attacks.

3.1.3. Correlation Coefficients. To better measure the correlation of adjacent pixels of an encrypted image, the correlation is frequently used to quantitatively describe its size. The specific definitions are as follows:

Input: encrypted image E and keys $x\text{Pre } d, y\text{Pre } d, z\text{Pre } d, x', y', z'$
Output: restored image reF
Step 1: $reR', reB', reG' = \text{reART}(E(1,2,3), n_2)$
Step 2: perform diffusion recovery.
  $\text{RD}_{M*N} = (0 \oplus y'_{M*N}) \oplus reR'_{M*N}$
  For $i = M * N - 1 : -1 : 1$
    $\text{RD}_i = (reR'_{i+1} \oplus y'_i) \oplus reR'_i$
  End
  $\text{RE}_1 = (0 \oplus x'_1) \oplus \text{RD}_1$
  For $i = 2 : M * N$
    $\text{RE}_i = (\text{RD}_{i-1} \oplus x'_i) \oplus RD_i$
  End
  $\text{RE} = \text{reshape}(\text{RE}, M, N)$
  .....
Step 3: $reE_{Im} = \text{reART}(\text{RE}, \text{RG}, \text{RB}, n_1)$
Step 4: $reF = \text{embedKey}(reE_{Im}, x\text{Pre } d, y\text{Pre } d, z\text{Pre } d)$

Algorithm 3: Pseudocode of the image decryption algorithm.

$$r_{pq} = \frac{|cov(p,q)|}{\sqrt{D(p)\sqrt{D(q)}}}. \tag{6}$$

In Formula (6),

$$\begin{cases} E(p) = \dfrac{1}{N}\sum_{i=1}^{N} p_i, \\ E(q) = \dfrac{1}{N}\sum_{i=1}^{N} q_i, \\ D(p) = \dfrac{1}{N}\sum_{i=1}^{N} (p_i - E(p))^2, \\ D(q) = \dfrac{1}{N}\sum_{i=1}^{N} (q_i - E(p))^2, \\ cov(p,q) = \dfrac{1}{N}\sum_{i=1}^{N} (p_i - E(p))(p_i - E(p)), \end{cases} \tag{7}$$

where p and q represent two vectors of adjacent pixels, and r_{pq} show the relational value. The lower the correlation of adjacent pixels, the closer their correlation is to 0.

Through simulation experiments (see Table 2), the correlation in all three directions is close to zero, indicating that the cryptographic algorithm proposed in this paper has almost no correlation.

3.1.4. Information Entropy. Information entropy is mainly used in the field of image processing to describe the information contained in an image, that is, the distribution of grayscale values in an image. Its formula is defined as

$$H(x) = -\sum_{i=1}^{n} p(x)\log_2 p(x_i), \tag{8}$$

where $p(x_i)$ represents the frequency with which grayscale x occurs.

The closer the value of the information entropy is to eight, the more uniform the distribution is. By comparing with other studies (see Table 3), it can be discovered that after employing the scheme proposed here, the encrypted image possesses a marked degree of randomness.

3.2. Robustness Analysis. As we all know, there may be various situations like information loss and attack in the process of image transmission [21]. A robust encryption scheme should be able to resist various common image attacks to a certain extent and should ensure the information missing or contaminated encrypted images can still obtain the main information of the original image through decryption.

3.2.1. Cropping Attack. Cropping attack refers to dividing the image, which causes the image yield some information. In image encryption, part of the encrypted image is cut and then decrypted, and the decrypted reconstructed image is compared with the original plaintext image [22].

We tested 25% and 50% cut attacks on three groups of experimental pictures with different resolutions and the test

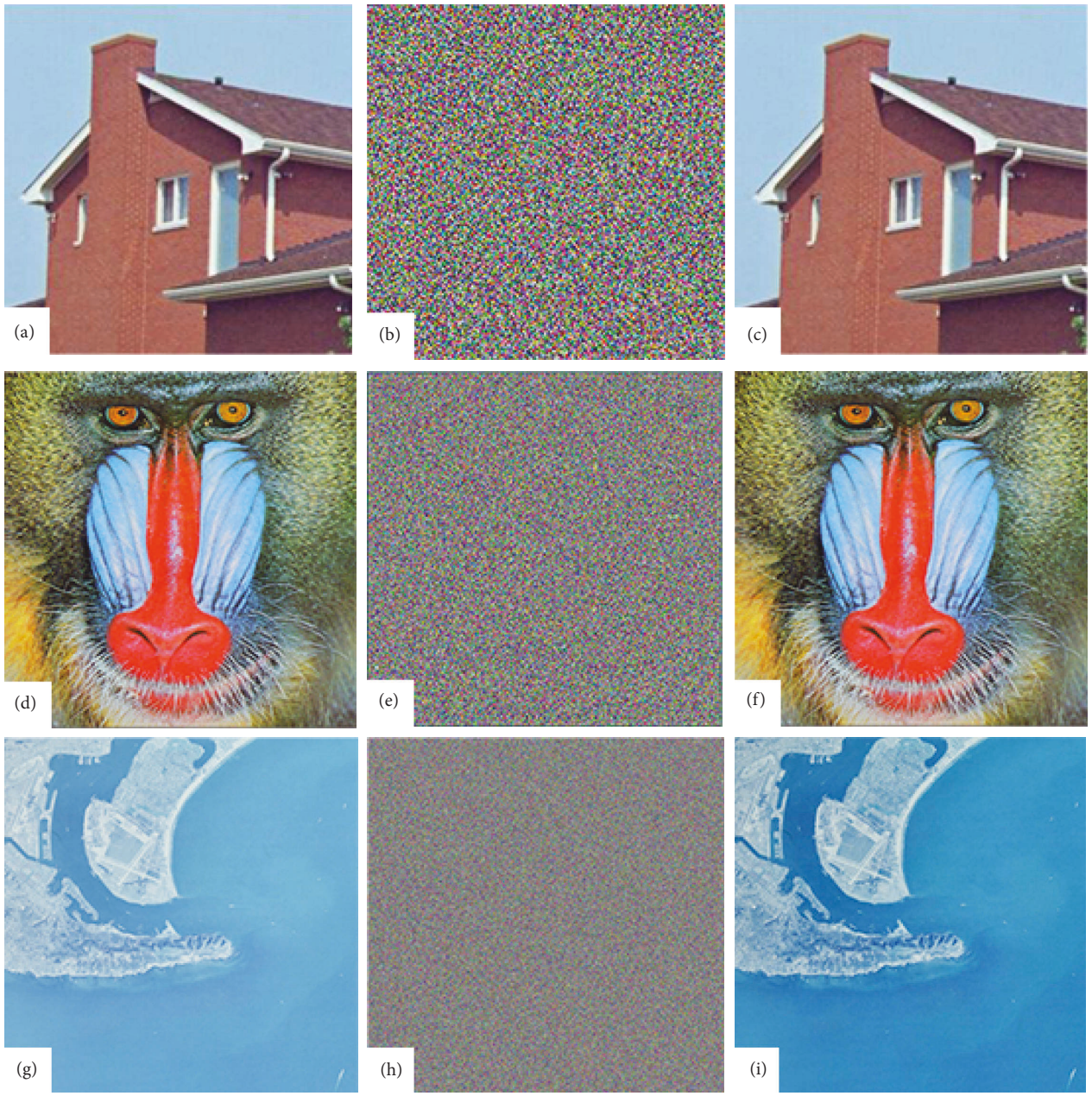

Figure 4: Experimental results of color image encryption and decryption. Image (a) is a color house original of 256 ∗ 256, (b) is an encrypted image of (a), and (c) is a decrypted image of (b). Image (d) is the original color monkey image of 512 ∗ 512, (e) is the encrypted image of (d), and (f) is the decrypted image of (e). Image (g) is an original color map of 1024 ∗ 1024, (h) is an encrypted image of (g), and (i) is a decrypted image of (h).

results (see Figure 7). From the experimental results, we can see that the encryption scheme can still display the main image information after the clipping attack which proves that the encryption scheme possesses good resistance to the clipping attack.

3.2.2. Gaussian Noise Attack. Images are always degraded by some senseless error, which is called noise. The ideal noise is considered as white noise, which appears at the same intensity at all frequencies. As a particular case of white noise, Gaussian noise can be used to approximate the noise in many real scenes.

In this paper, three groups of experimental images are decrypted by adding 0.08 and 0.2 Gaussian white noise, respectively. The experimental results are shown in Figure 8. It can be noted from the diagram that the resulting diagram encrypted by this encryption scheme can obtain the main information of the original image through decryption even if it is polluted by noise, which has good robustness to noise attack.

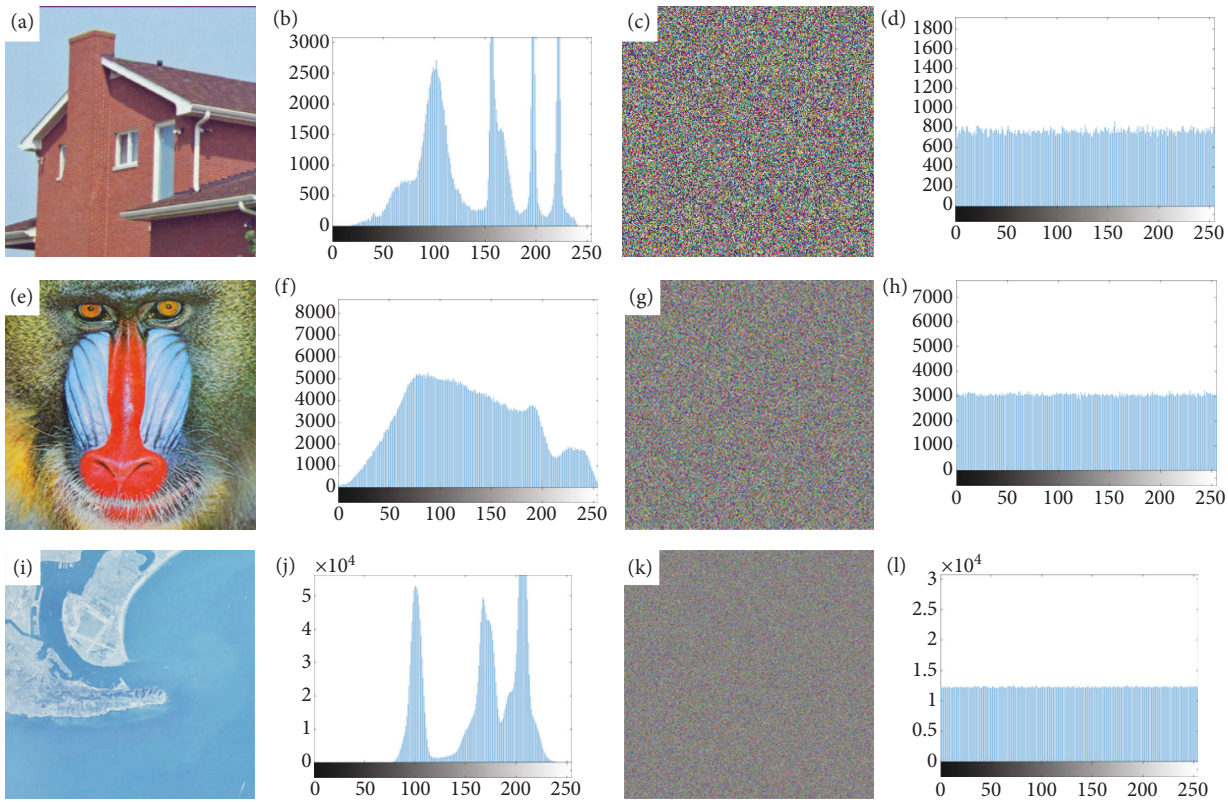

Figure 5: Color image encryption and decryption histogram. Image (a) is displayed as a color house image, image (b) is a histogram of (a), image (c) is an encrypted image of (a), and image (d) is a histogram of (c). Image (e) is displayed as a color monkey image, image (f) is a histogram of (e), image (g) is an encrypted image of (e), and image (h) is a histogram of (g). Image (i) is a color map image, image (j) is a histogram of (i), image (k) is an encrypted graph of (i), and image (l) is a histogram of (k).

3.3. Encryption Performance Analysis

3.3.1. Peak Signal-to-Noise Ratio. The peak signal-to-noise ratio (PSNR) measures the maximum signal-to-noise ratio on a signal and is typically used for image and video signals. PSNR is calculated using the following formula:

$$\mathrm{PSNR}\,(\mathrm{dB}) = N \times \frac{\mathrm{MAX}^2}{\sum_{n=1}^{N} (X_n - Y_n)^2}, \tag{9}$$

where N is the total number of samples in the signal, MAX is the maximum possible value of the sample, X_n corresponds to the nth sample of the original signal X, and Y_n represents the nth sample of the encrypted signal Y.

PSNR is a measure of the difference in peak error between two images. For ideally similar images, PSNR is infinite, and for completely different images, its value is zero. The PSNR of the images was encrypted implementing the algorithm proposed in this paper and is calculated together with the original images (see Table 4), and it is found that their values are lower than the PSNR values calculated by other research schemes, which proves that the images encrypted by the algorithm in this paper are quite different from the original images, and the algorithm has good encryption performance.

3.3.2. Key Space Analysis. The space size of the encryption key affects the encryption performance of the encryption scheme. A robust image encryption scheme has a requirement of a sufficiently large key space [26]. It can be seen from the above that in the encryption scheme of this paper, we got six encryption keys. When $M = N$, we obtain the length of $x\mathrm{Pred}, y\mathrm{Pred}, z\mathrm{Pred}$, respectively:

$$\mathrm{len}\,(x\mathrm{Pred}, y\mathrm{Pred}, z\mathrm{Pred}) = M - \mathrm{mod}\,(M, 10). \tag{10}$$

Meanwhile, the key length of x', y', z' can be expressed as

$$\mathrm{len}\,(x', y', z') = \left[\mathrm{ceil}\left(\frac{M * N}{1000}\right) * 1000 + 1\right]. \tag{11}$$

From what has been mentioned above, we can conclude that the key space size is

$$[M - \mathrm{mod}\,(M, 10)]^3 * \left[\mathrm{ceil}\left(\frac{M * N}{1000}\right) * 1000 + 1\right]^3. \tag{12}$$

According to formula (12), we calculate the key space of three groups of experimental graphs, and the results are

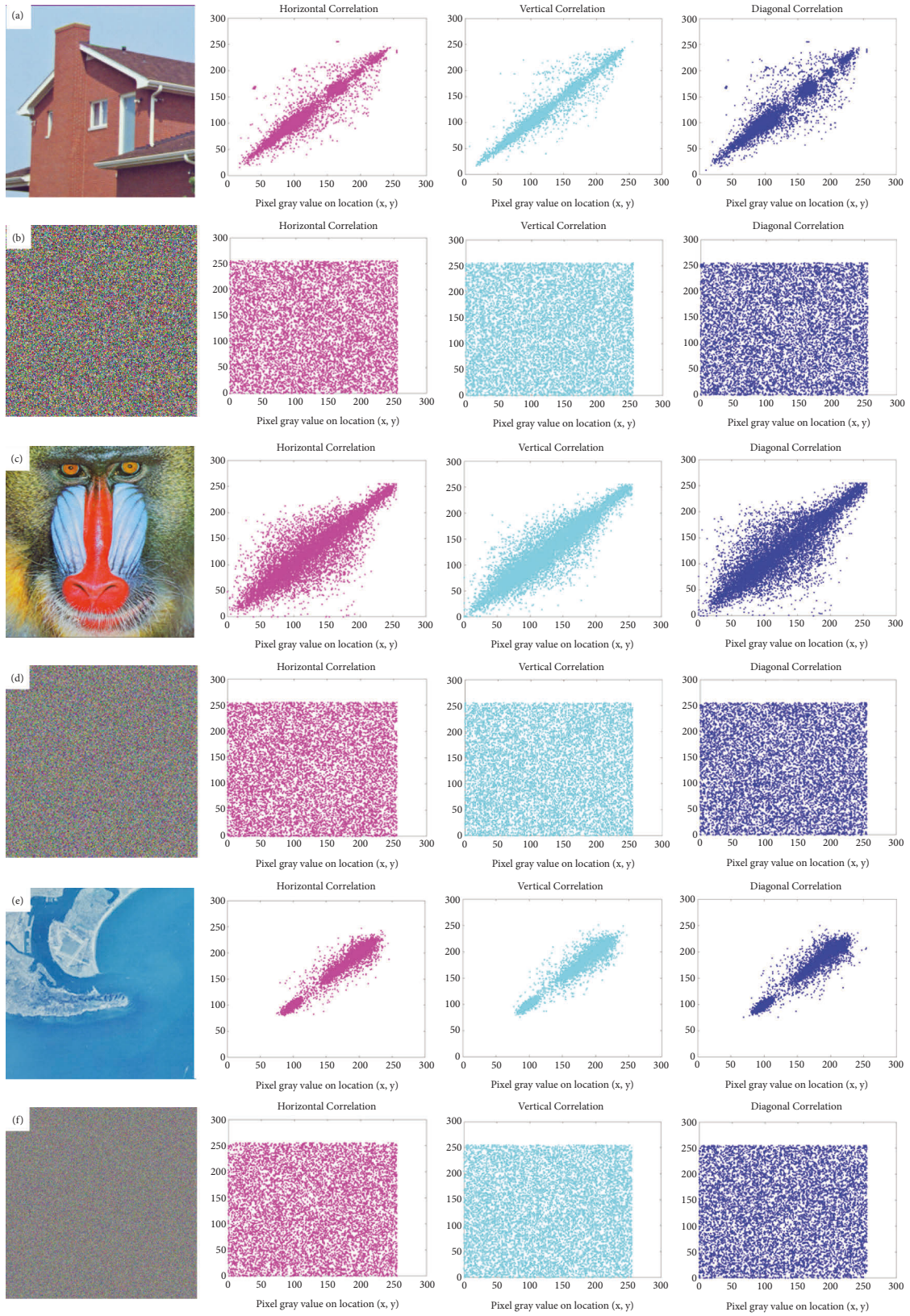

Figure 6: Correlation analysis of related pixels of three groups of experimental diagrams. Image (a) is a color house image, and the correlation analysis diagram of "horizontal, vertical and diagonal" adjacent pixels of (a) from left to right. The image (b) is an encrypted image of (a), and the correlation analysis images of adjacent pixels in three directions of (b) are sequentially from left to right. Image (c) is a color monkey image, and the correlation analysis diagram of adjacent pixels in three directions from left to right is (c). The image (d) is an encrypted image of (c), and the correlation analysis images of adjacent pixels in three directions of (d) are sequentially from left to right. The image (e) is a color map image, and the correlation analysis graphs of adjacent pixels in three directions of (e) are arranged from left to right. Image (f) is the encrypted graph of (e) and is the correlation analysis graph of adjacent pixels in the three directions of (f) from left to right.

Table 2: Correlation coefficients of the encrypted Lena image for different methods.

Method	Components	Direction		
		Horizontal	Vertical	Diagonal
Plain image	R	0.9724	0.9429	0.9241
	G	0.9727	0.9468	0.9250
	B	0.9463	0.9012	0.8638
Proposed	R	−0.0046	0.0072	0.0009
	G	−0.0015	0.0056	−0.0125
	B	−0.0091	−0.0076	−0.0145
Ref. [10]	R	0.0064	0.0160	−0.0026
	G	0.0009	0.0034	0.0125
	B	0.0091	−0.0045	−0.0090
Ref. [18]	R	−0.0112	−0.0026	0.0052
	G	0.0050	0.0199	−0.0064
	B	−0.0179	0.0120	−0.0161
Ref. [19]	R	0.0014	0.0048	0.0002
	G	0.0033	−0.0006	0.0048
	B	0.0021	0.0002	−0.0040
Ref. [20]	R	−0.0070	0.0049	−0.0083
	G	−0.0076	0.0062	0.0032
	B	0.0047	−0.0025	−0.0058

Table 3: Information entropy of the encrypted Lena image for different methods.

Method	Red	Green	Blue	Average
Proposed	7.9916	7.9913	7.9919	7.9916
Ref. [10]	7.9974	7.9976	7.9974	7.9975
Ref. [18]	7.9895	7.9894	7.9894	7.9894
Ref. [19]	7.9917	7.9912	7.9918	7.9916
Ref. [20]	7.9991	7.9993	7.9993	7.9992

$4.4923 * 10^{21}, 2.4131 * 10^{24}, 1.225 * 10^{27}$ that are all greater than 2^{70}. Experimental results show that this encryption scheme has large key space and high security.

3.3.3. Time Complexity. Time complexity, as a function, is consumed to qualitatively describe the running time of the algorithm and is an important indicator to measure the quality of the algorithm. The common letter O is employed to denote and does not include the low-order term and the first-term coefficient of the function.

The algorithm proposed in this paper mainly encrypts images with equal lengths and widths, assuming the length and width of the images represent n. The chaotic system is operated to generate the initial key sequence as needed $1/1000n^2$. At the same moment, we estimate the time complexity of the deep learning model as 1 based on its training theory. The proposed embedding key method in this paper requiresn^2. Scrambling and diffusion algorithms are required in the schemes $3xn^2\,(x \in [0, 255])$ and $6n^2$, respectively.

Composing the previous estimates, we can conclude that the time complexity function formula for the cryptographic algorithm proposed in this article is

$$\frac{1001\,(7 + 3x)}{1000}n^2 + 1. \tag{13}$$

Formula (13) is simplified by order of magnitude to give the time complexity of the cryptographic algorithm presented $O(n^2)$ herein.

4. Discussion

Through the observation of the above simulation experimental results, it can be perceived that the cryptographic algorithm proposed in this paper performs well in terms of histogram, correlation, information entropy, and other statistical performance, and it shows good robustness in common shearing and noise attacks. As a group, it obtains an enormous key space and lower time complexity and better encryption performance such as the PSNR value compared with other research schemes.

Because the encryption algorithm proposed in this paper is primarily to operate a chaotic system combined with a deep learning model to generate an encryption key, and then embed the key through an embedding method designed by ourselves. In theory, deep learning models may generate errors when generating keys. But this article focuses on improving the algorithmic security of color image encryption, so the possible errors are unconsidered in this encryption scheme. Naturally, any algorithm includes certain limitations. In the effective experimental process, we tested the pixel number change rate (NPCR) and the uniform

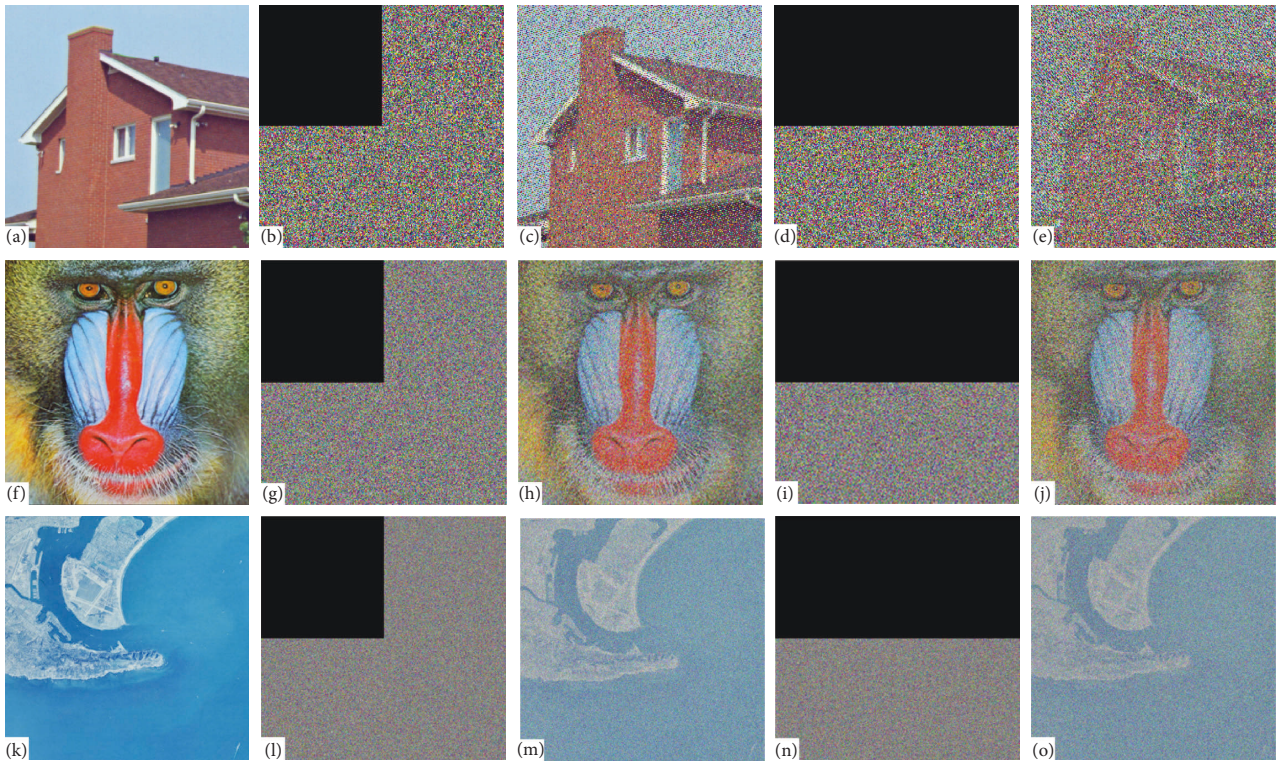

Figure 7: Cropping attack analysis diagram. Image (a) is color house image, (b) is a result map of 25% clipping of the encrypted map (a), (c) is a decryption map of (b), (d) is a result map of 50% clipping of the encrypted map (a), and (e) is a decryption map of (d). Image (f) is color monkey original image, (g) is a result image of 25% cutting of the encrypted image (f), (h) is a decryption image of (g), (i) is a result image of 50% cutting of the encrypted image (f), and (j) is a decryption image of (i). The image (k) is color map image, (l) is a result map of 25% cropping of the encrypted map (k), (m) is a decryption map of (l), (n) is a result map of 50% cropping of the encrypted map (k), and (o) is a decryption map of (n).

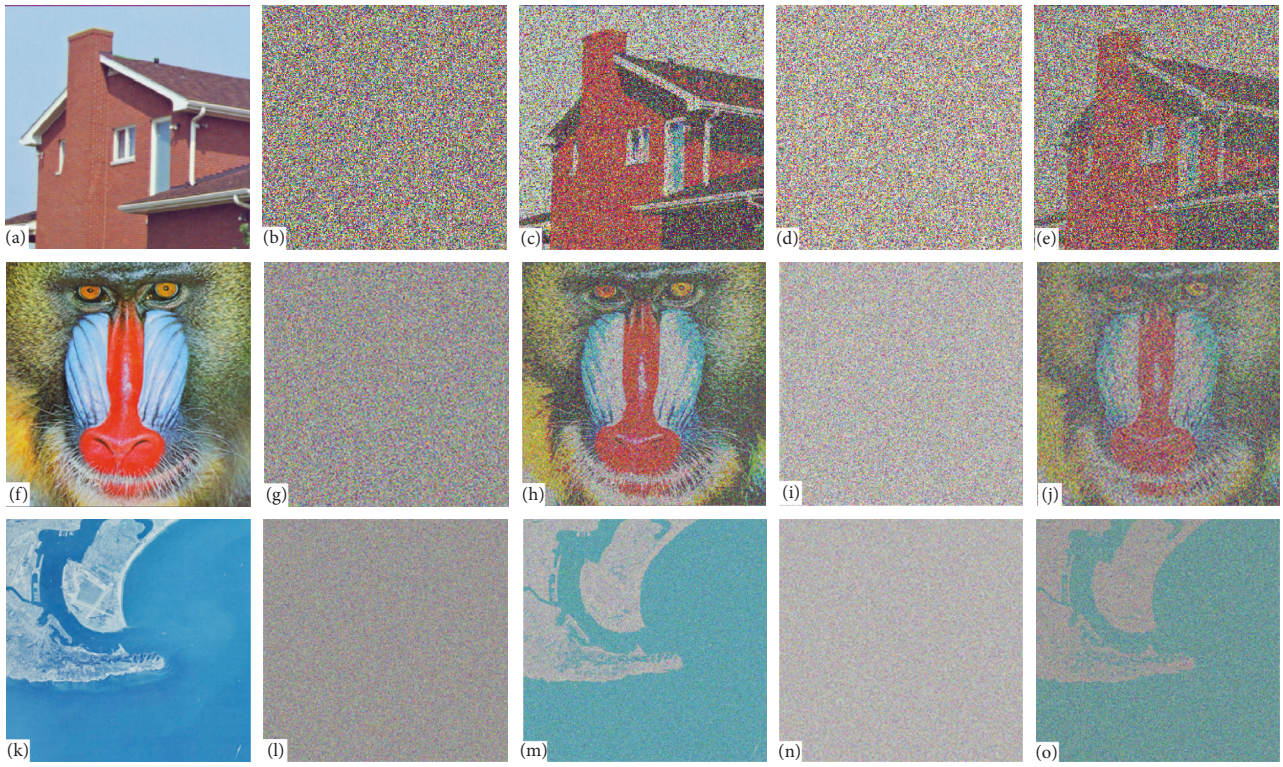

Figure 8: Gaussian noise attack analysis diagram. Image (a) is color house image, (b) is a result image of adding density 0.08 Gaussian noise to the encrypted image (a), (c) is a decryption image of (b), (d) is a result image of adding density 0.2 Gaussian noise to the encrypted image (a), and (e) is a decryption image of (d). Image (f) is color monkey image, (g) is a result image of adding density 0.08 Gaussian noise to (f), (h) is a decryption image of (g), (i) is a result image of adding density 0.2 Gaussian noise to (f), and (j) is a decryption image of (i). The image (k) is color map image, (l) is a result image of adding a density of 0.08 Gaussian noise to the encrypted image (k), (m) is a decryption image of (l), (n) is a result image of adding a density of 0.2 Gaussian noise to the encrypted image (k), and (o) is a decryption image of (n).

Table 4: PSNR of the Lena image for different methods.

Method	Proposed	Ref. [23]	Ref. [24]	Ref. [25]
PSNR	7.1954	7.6324	8.6258	7.6714

means change intensity (UACI). It is found that the resistance to differential attack was weaker than that of other research results and temporarily only supports the encryption of color images of equal length and width.

5. Conclusion

In this paper, we proposed a color image encryption scheme based on deep learning and block embedding. Because of its multilevel complex structure, the deep learning model is combined with a chaotic system to ensure the complexity of the generated key. As a group, we design a block embedding method, which combines the encrypted key with the color image organically. Through simulation experiments, it is proved that the proposed encryption scheme obtains good performance and robustness to general attacks.

MATLAB uses dual-precision data types by default. To overcome the difference between the simulation accuracy and the factual situation in the encryption algorithm design, we reduce the frequency of the conversion of different data types in the experiment. Meantime, the complex calculation that generates the encryption key is separated from the operation of the key embedding to reduce the complex operation of the color image itself. As one would expect, this paper temporarily only realizes the color image encryption of equal length and width. In the future, we will study the color image encryption with unusual length and width in this direction and improve the robustness of the algorithm to other image attack methods.

Data Availability

The data that support the findings of this study are openly available at [https://sipi.usc.edu/database/].

Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents conflicts of interest in connection with the work submitted.

References

[1] Q. Xu, K. Sun, C. Cao, and C. Zhu, "A fast image encryption algorithm based on compressive sensing and hyperchaotic map," *Optics and Lasers in Engineering*, vol. 121, pp. 203–214, 2019.

[2] X. Li, Y. Jiang, M. Chen, and F Li, "Research on iris image encryption based on deep learning," *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, p. 126, 2018.

[3] J. Chen, X. W. Li, and Q. H. Wang, "Deep learning for improving the robustness of image encryption," *IEEE Access*, vol. 7, pp. 181083–181091, 2019.

[4] R. Ni, F. Wang, J. Wang, and Y Hu, "Multi-image encryption based on compressed sensing and deep learning in optical gyrator domain," *IEEE Photonics Journal*, vol. 13, no. 3, pp. 1–16, 2021.

[5] Q. Zhang and J. Li, "Single exposure phase-only optical image encryption and hiding method via deep learning," *IEEE Photonics Journal*, vol. 14, no. 1, pp. 1–8, Article ID 7813508, 2022.

[6] Z.-P. Zhao, S. Zhou, and X.-Y. Wang, "A new chaotic signal based on deep learning and its application in image encryption," *Acta Physica Sinica*, vol. 70, no. 23, Article ID 230502, 2021.

[7] G. Chen and T. Ueta, "Yet another chaotic attractor[J]," *International Journal of Bifurcation and chaos*, vol. 09, no. 07, pp. 1465-1466, 1999.

[8] S. Siami-Namini, N. Tavakoli, and A. S. Namin, *The Performance of LSTM and BiLSTM in Forecasting Time series*, IEEE, in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292, IEEE, May 2019.

[9] S. Liu, C. Guo, and J. T. Sheridan, "A review of optical image encryption techniques," *Optics & Laser Technology*, vol. 57, pp. 327–342, 2014.

[10] K. M. Hosny, S. T. Kamal, and M. M. Darwish, *A Color Image Encryption Technique Using Block Scrambling and chaos*, Multimedia Tools and Applications, 2021.

[11] X. Wang, N. Guan, and J. Yang, "Image encryption algorithm with random scrambling based on one-dimensional logistic self-embedding chaotic map," *Chaos, Solitons & Fractals*, vol. 150, no. 3, Article ID 111117, 2021.

[12] M. Younes and J. Aman, "Image encryption using block-based transformation algorithm[J]," *IAENG International Journal of Computer Science*, vol. 35, 2011.

[13] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H Arshad, "State-of-the-art in artificial neural network applications: a survey," *Heliyon*, vol. 4, no. 11, Article ID e00938, 2018.

[14] O. I. Abiodun, M. U. Kiru, A. Jantan et al., "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158820–158846, 2019.

[15] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos, Solitons & Fractals*, vol. 139, Article ID 110045, 2020.

[16] Y. Sang, J. Sang, and M. S. Alam, "Image encryption based on logistic chaotic systems and deep autoencoder," *Pattern Recognition Letters*, vol. 153, pp. 59–66, 2022.

[17] A. V. Diaconu and A. C. Dascalescu, "Correlation distribution of adjacent pixels randomness test for image encryption[C]," *Proc. Rom. Acad. Ser. A*.vol. 18, pp. 351–360, 2017.

[18] X. Wu, J. Kurths, and H. Kan, "A robust and lossless DNA encryption scheme for color images," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12349–12376, 2018.

[19] Y. Q. Zhang, Y. He, P. Li, and X. Y. Wang, "A new color image encryption scheme based on 2DNLCML system and genetic operations," *Optics and Lasers in Engineering*, vol. 128, no. 3, Article ID 106040, 2020.

[20] G. Cheng, C. Wang, and H. Chen, "A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture," *International Journal of Bifurcation and Chaos*, vol. 29, no. 09, Article ID 1950115, 2019.

[21] M. N. Abdulwahed and A. K. Ahmed, "Improved anti-noise attack ability of image encryption algorithm using de-noising technique," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 3080–3087, 2020.

[22] Q. Chen, X. Shen, Y. Cheng, C. Lin, Y. Liu, and B. Zhou, "A security-enhanced joint transform correlator optical encryption system with cropping operation," *Optik*, vol. 245, Article ID 167654, 2021.

[23] B. Harjo and D. Setiadi, "Improved color image encryption using hybrid modulus substitution cipher and chaotic method," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, pp. 157–166, 2021.

[24] F. Budiman and D. Setiadi, "A combination of block-based chaos with dynamic iteration pattern and stream cipher for color image encryption," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 6, pp. 131–141, 2020.

[25] L. Liu, L. Zhang, D. Jiang, Y. Guan, and Z Zhang, "A simultaneous scrambling and diffusion color image encryption algorithm based on hopfield chaotic neural network," *IEEE Access*, vol. 7, no. 99, pp. 185796–185810, 2019.

[26] J. Zheng and L. F. Liu, "Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map," *IET Image Processing*, vol. 14, no. 11, pp. 2310–2320, 2020.