Aiming at the problems of small key space, low security, and low algorithm complexity in a low-dimensional chaotic system encryption algorithm, an image encryption algorithm based on the ML neuron model and DNA dynamic coding is proposed. The algorithm first performs block processing on the R, G, and B components of the plaintext image to obtain three matrices, and then constructs a random matrix with the same size as the image components through logistic mapping and performs DNA encoding, DNA operation, and DNA decoding on the two parts. Second, it performs determinant permutation on the matrix by two different chaotic sequences obtained by logistic mapping iteration. Finally, it merges the block and image components to complete the image encryption and obtain the ciphertext image. Wherein, DNA encoding, DNA operation, and DNA decoding methods are all randomly and dynamically determined by the chaotic sequence generated by the ML neuron chaotic system. According to simulation results and performance analysis, the algorithm has a larger key space, can effectively resist various statistical and differential attacks, and has better security and higher complexity.

1. Introduction

With the rapid development of cloud computing technology [1] and the widespread application of the 5G network technology [2] in recent years, digital image transmission has already become an indispensable part in all walks of life. As digital images contain a large amount of important data such as security privacy and confidential information, there is a risk of theft in the transmission of digital image information in various fields. Fortunately, the birth of image encryption technology [3–6] solves this problem and ensures the confidentiality of digital images. Since chaotic systems have the characteristics of sensitivity to initial values, pseudo-randomness, and unpredictability, more and more image encryption algorithms based on chaos have been proposed [7–9]. The image encryption algorithm proposed in literature [7] is based on the image segmentation and a multi-diffusion model, which improves the security of the algorithm. The chaotic image encryption algorithm proposed in the literature [8] combines two methods of block scrambling and dynamic index diffusion, and its advantages are fast and safe. The image encryption algorithm based on the fractional order chaotic system proposed in literature [9] has higher security and execution efficiency.

The advantage of DNA computing lies in that it can perform parallel computing, which reduces power consumption and has a high storage density. Researchers have explored the complementary base pairing principle in DNA theory [10–12] and introduced it to realize image encryption. Literatures [13, 14] proposed many DNA-based encryption schemes. Although these schemes combine the chaotic system with DNA encoding theory, which improves the complexity of the system, the security performance still needs improvement because of the not-high chaotic dimension and the not-large key space. In recent years, more and more new image encryption algorithms that combine chaotic mapping and DNA encoding have been proposed by scholars. The newly emerged algorithms have better encryption effects and are more secure than previous ones. Therefore, many researchers from various fields have also begun to pay attention to and deeply understand this type of
an image encryption algorithm that combines chaotic mapping and DNA encoding [15–18]. This paper proposes a new image encryption algorithm to generate the required ciphertext image more securely by adopting the DNA random dynamic coding method and combining the ML neuron chaotic system.

### 2. Related Theoretical Knowledge

#### 2.1. ML Neuron Model

The ML neuron model is a two-dimensional model [19–21] based on the research results of the muscle fibers of the Arctic goose. This article uses an improved three-dimensional ML neuron model [22, 23], and its differential equation is as follows:

\[
\begin{align*}
\frac{dV}{dt} & = -u - g_l(V - V_l) - g_k\omega(V - V_k) - g_{ca}m_{\infty}(V - V_{ca}), \\
\frac{d\omega}{dt} & = \lambda(V)(\omega_{\infty}(V) - \omega), \\
\frac{du}{dt} & = \mu(0.2 + V).
\end{align*}
\]

(1)

In formula (1): \(V\) is the neuron membrane potential; \(u\) is the neuron recovery variable; \(\omega\) is the slow-changing regulation current; \(g_k, g_{ca}\), and \(g_l\) are the maximum conductance of \(K^+\), \(Ca^{2+}\), and the leakage current channel, respectively; \(V_k, V_{ca}\), and \(V_l\) are the reversal voltages corresponding to the abovementioned channels, respectively; \(\lambda(V)\) is the activation time constant; \(\omega_{\infty}(V)\) and \(m_{\infty}(V)\) are the steady-state values of the opening probability of the ion channel \(K^+\) and \(Ca^{2+}\), respectively; and \(\mu\) is a constant.

Part of the parameters of the abovementioned ML neuron model are set, respectively, as \(g_{ca} = 1.2 \mu F/cm^2\), \(V_{ca} = 1 \text{ mV}\), \(g_k = 2 \mu F/cm^2\), \(g_l = 0.5 \mu F/cm^2\), \(V_l = -0.5 \text{ mV}\), \(\mu = 0.005\), and keep other parameters unchanged. When the parameter \(V_{ca} = 0.83 \text{ mV}\), the ML neuron model is at the
When state, and its dynamic behavior is more complicated. The logistic equation is also called the insect-population model [24, 25], and its equation is as formula (2):

\[ x(n+1) = \mu x(n) [1 - x(n)] \]

wherein \( \mu \) is the system parameter and \( x(n) \in (0, 1) \), \( \mu \in (0, 4) \). When \( \mu \in (3.569945, 4) \), logistic mapping is in a chaotic state, and its dynamic behavior is more complicated.

### 2.2. Logistic Chaotic Mapping

The logistic equation is also called the insect-population model [24, 25], and its equation is as formula (2):

\[ x(n+1) = \mu x(n) [1 - x(n)] \]

wherein \( \mu \) is the system parameter and \( x(n) \in (0, 1) \), \( \mu \in (0, 4) \). When \( \mu \in (3.569945, 4) \), logistic mapping is in a chaotic state, and its dynamic behavior is more complicated.

### Figure 3: Flowchart of the image encryption algorithm.

![Flowchart of the image encryption algorithm](image)

**Input:** Original plaintext image \( I \)

1. Divide the original plaintext image \( I \) into three two-dimensional matrices.
2. Divide the three two-dimensional matrices into blocks of size \( l \times l \), fill the three two-dimensional matrices so that their sizes all satisfy Algorithm 1:
   \[
   \begin{align*}
   \text{mod}(M, t) &= 0 \\
   \text{mod}(N, t) &= 0
   \end{align*}
   \]
3. Set the initial value \( x_0 \) according to Algorithm 1, and obtain the chaotic sequence after successive iterations of the Logistic chaotic mapping:
   \[ x_i = \sum I_i (x, y) / (255 \times M \times N) \quad (i = 1, 2, 3) \]
4. Process the sequence according to Algorithm 1 so that all elements are belong to \([0, 255]\), and divide it into random matrices of the same size \( l \times l \), \( p_j = \text{mod} \left( \text{cell} \left( p_j \times 10^3 \right), 256 \right) \).
5. Use MATLAB’s built-in Runge–Kutta function \text{ode45} to calculate to obtain the three sequences of length \((M \times N)/2\) in the ML neuron model. The initial value is calculated by Algorithm 1:
   \[
   \begin{align*}
   X(0) &= \sum (\text{bi tan} \left( I_j (17) \right)) / (17 \times M \times N) \\
   Y(0) &= \sum (\text{bi tan} \left( I_j (34) \right)) / (34 \times M \times N) \\
   Z(0) &= \sum (\text{bi tan} \left( I_j (68) \right)) / (68 \times M \times N)
   \end{align*}
   \]
6. According to the initial value and the number of blocks \( r \), solve the ML neuron model and get four chaotic sequences \( X, Y, Z, \) and \( H \).
7. Use \( X \) and \( Y \) to determine the DNA encoding method of the image matrix and the chaotic matrix. There are 8 kinds, 1–8.
8. Use \( Z \) to determine the DNA operation method between the image matrix and the chaotic matrix. There are 4 kinds, 0–3.
9. Use \( H \) to perform DNA decoding operations on the computed sub-blocks. There are 8 kinds, 1–8.
10. Obtain two Logistic chaotic sequences. The initial value is obtained according to Algorithm 1 to produce two chaotic sequences:
   \[
   \begin{align*}
   x_{11} &= \sum (I_i (\cdot)) / 255 \times M \times N \times 2 \\
   x_{12} &= \sum (I_i (\cdot)) / 255 \times M \times N \times 2
   \end{align*}
   \]
11. Sort the two chaotic sequences obtained in the previous step in descending order and obtain the position sequence before the ranking. Take the sequence value before ranking and its corresponding index as the exchange coordinates of row and column to perform row and column permutation on the matrix.
12. Combine the three component matrices after row and column permutation.

**Output:** Ciphertext image

**Algorithm 1:** The image encryption algorithm.

---

2.3. DNA Encoding and Operations

2.3.1. DNA Encoding. There are 4 kinds of nucleotides in a DNA sequence, namely, A (adenine), G (guanine), C (cytosine), and T (thymine). Wherein, A and T, G, and C have complementary relationships, which are very similar to that of the binary code 0 and 1 in mathematics, so 00 and 11, 01 and 10 also have a complementary relationship. A, T, G, and C are represented by two-digit binary 00, 11, 10, and 01, respectively. According to the principle of complementary pairing, there are 8 sets of encoding rules. As shown in Table 1, the corresponding DNA decoding rules and DNA encoding rules are on the contrary.
2.3.2. DNA Operations. In order to make the algorithm more complex, this paper uses 4 kinds of DNA operations at the same time, namely, DNA addition, DNA subtraction, DNA XOR, and DNA XNOR operations. The operation rules are similar to those of addition, subtraction, XOR, and XNOR in binary mathematics. Because there are 8 DNA encoding methods, so there are 8 corresponding operation rules. Taking one of the DNA encoding methods as an example, the DNA addition, subtraction, XOR, and XNOR operations are shown in Tables 2–5.

3. Image Encryption Algorithm

This paper divides the color digital image into three two-dimensional matrices, which greatly reduce the time and space resources required. DNA encoding and calculation are performed on each two-dimensional matrix block, then row and column permutations are performed again after encryption, and finally the three channels are merged to obtain a color encrypted image. The three chaotic sequences generated by the ML neuron chaotic system determine the DNA encoding, decoding, and operation rules of each block. The initial value of the chaotic system is determined by the original image to ensure the effect of “one image, one cryptogram.” Three different chaotic sequences which are obtained by performing three iterations on the logistic mapping are used to perform DNA operations with the original image, as well as row and column permutations after DNA decoding operations. The flowchart of the image encryption algorithm is shown in Figure 3. The specific encryption procedure can be summed up by Algorithm 1.

The decryption algorithm is the inverse process of the abovementioned encryption algorithm and will not be described in detail here.

4. Experimental Simulation and Safety Performance Analysis

4.1. Experimental Simulation. The experiment was completed on the MATLABR2017, a platform in the Win10 system, and the algorithm in this paper was used to encrypt and decrypt a color Lena image with a size of 512 \times 512. The result of encrypting and decrypting the Lena image is shown in Figure 4.

4.2. Safety Performance Analysis

4.2.1. Histogram Analysis. The grayscale histogram of the image can objectively reflect the distribution of image pixels. Figure 5 shows the grayscale histogram of the R, G, and B channels before and after the Lena image encryption through the comparison of grayscale.

In order to quantitatively compare whether the difference between the histograms of the test ciphertext and the ideal ciphertext image is significant, a chi-square test is used to analyze the uniformity of the histograms. The calculation method is shown in formula (3):

$$
\chi^2 = \sum_{i=1}^{256} \frac{(f_i - f)^2}{f}
$$

wherein, $f = M \times N/256$, $M$, $N$ represents the size of the image. $f_i$ is the frequency of occurrence of gray value $i$.

When the significance level is 0.05, the corresponding chi-square critical value is 293.2478. The calculation results of the chi-square analysis are shown in Table 6, from which it can be seen that the chi-square values of all ciphertexts are less than 293.2478, and the encryption algorithm has passed the chi-square test. Therefore, the encryption algorithm proposed in this paper can effectively resist statistical attacks [26].

4.2.2. Correlation Analysis of Adjacent Pixels. The correlation between adjacent pixels of a plaintext image is very strong, so the image is vulnerable to statistical analysis attacks. The calculation formula of the correlation coefficient between two pixels in the image is as shown in formulas (4)–(7):
Figure 5: Comparison of histograms before and after image encryption. (a) R channel histogram of the image before encryption; (b) R channel histogram of the image after encryption; (c) G channel histogram of the image before encryption; (d) G channel histogram of the image after encryption; (e) B channel histogram of the image before encryption; and (f) B channel histogram of the image after encryption.
value, which shows that the randomness of the ciphertext condition, the information entropy will be at the maximum values. When the distribution probability reachesthe equal probability, the information entropy referstothedistribution probability of image gray pixels. Different selection methods of x, y will lead to the correlation coefficients in different directions, including the horizontal direction, the vertical direction, and the opposite angle direction.

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i, \\
D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \\
cov(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)), \\
r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)},}
\]

wherein \( N \) is the logarithm of the required pixel, and x, y are the gray values of adjacent pixels. Different selection methods of x, y will lead to the correlation coefficients in different directions, including the horizontal direction, the vertical direction, and the opposite angle direction.

\( N \) pixels point pairs are randomly selected from different directions, which makes \( N = 5000 \), and the correlation coefficient of each plaintext and ciphertext is calculated. It can be seen from the results of a certain experiment shown in Table 7 that the correlation coefficients of the three plaintext images in each direction are very large and close to 1, indicating that the adjacent pixels in each direction of the plaintext image are extremely correlated; at the same time, the correlation coefficients of the three ciphertext images are all extremely small and very close to 0, indicating that the correlation between adjacent pixels in all directions of the ciphertext image is extremely weak. In addition, compared with the methods proposed in literature [27] and literature [28], the algorithm proposed in this paper can eliminate the correlation between adjacent pixels and conceal the data characteristics of the original image.

Figures 6–8 are the pixel distribution diagrams of three channels R, G, and B of the Lena plaintext and ciphertext image in different directions. It can be seen that the distribution of plaintext pixels is linear, while the distribution of ciphertext pixels is disorderly and irregular. Therefore, the algorithm in this paper can effectively reduce the correlation between adjacent pixels of plaintext images and can effectively resist statistical analysis attacks.

### 4.2.3. Global Entropy and Local Entropy Analysis

The definition of information entropy refers to the uncertainty of information, while the definition of image information entropy refers to the distribution probability of image gray values. When the distribution probability reaches the equal condition, the information entropy will be at the maximum value, which shows that the randomness of the ciphertext image is very high and that the encryption algorithm has a strong ability to resist statistical attacks. The calculation formula of global entropy is as formula (8):

\[
H(m) = - \sum_{i=1}^{L} p(m_i) \log_2 p(m_i),
\]

where \( L \) refers to the gray level of the image and \( p(m_i) \) represents the probability that the gray value \( m_i \) appears.

Table 7 shows the comparison with the encryption algorithms in existing literature. It can be seen that the global entropy of the encryption algorithm proposed in this paper is closer to the ideal value of 8.

On the basis of global entropy, Wu Y et al. proposed a calculation method called “local Shannon entropy,” which overcomes the shortcomings of inaccuracy, inconsistency, and low efficiency of global entropy [31]. The local entropy is an improvement over the global entropy. It first randomly selects non-overlapping blocks in the image and then calculates the average value of the global entropy for the small image blocks. According to the method described in the literature, 30 non-overlapping blocks are randomly selected in the image. When the number of pixels in each block is 1936, the local entropy is calculated, and the ideal value is 7.9024. Within the 0.01 level, if the local entropy is between 7.9017 and 7.9032, it means that the local entropy of the image is ideal and the image has good randomness. The local entropy test results are shown in Table 9.

### 4.2.4. Key Sensitivity Analysis

Key sensitivity refers to the resulting chaotic sequence changes after a small change to the key in the encryption algorithm, which has a great impact on the encrypted and decrypted image. If the key is not sensitive enough to correctly reconstruct the original image even with a slightly different key, the key may degenerate. In this paper, the key sensitivity test method of the encryption algorithm is given.

The first method is direct observation. The same encrypted image is decrypted with a slightly changed key and the differences between the images are observed. We can improve the anti-cracking ability of the encryption algorithm by increasing a certain number of digits after the decimal point of the key. The decrypted image obtained when one of the encrypting keys \( x_0 \) is changed from 0.5475 to 0.5476...
Figure 6: Adjacent pixels correlation of the R channel before and after encryption. (a) Horizontal correlation of the original image R channel; (b) vertical correlation of the original image R channel; (c) diagonal correlation of the original image R channel; (d) horizontal correlation of the ciphertext image R channel; (e) vertical correlation of the ciphertext image R channel; and (f) diagonal correlation of the ciphertext image R channel.
Figure 7: Adjacent pixels correlation of the G channel before and after encryption. (a) Horizontal correlation of the original image G channel; (b) vertical correlation of the original image G channel; (c) diagonal correlation of the original image G channel; (d) horizontal correlation of the ciphertext image G channel; (e) vertical correlation of the ciphertext image G channel; and (f) diagonal correlation of the ciphertext image G channel.
Figure 8: Adjacent pixels correlation of the B channel before and after encryption. (a) Horizontal correlation of the original image B channel; (b) vertical correlation of the original image B channel; (c) diagonal correlation of the original image B channel; (d) horizontal correlation of the ciphertext image B channel; (e) vertical correlation of the ciphertext image B channel; and (f) diagonal correlation of the ciphertext image B channel.
to 0.547500000000001 is shown in Figure 9. It can be found that the decryption is not successful when the key is changed slightly, which shows that the proposed encryption algorithm has good key sensitivity and can effectively resist the exhaustive attacks.

When one of the keys changes slightly, the histograms of the three channels of R, G, and B of the incorrectly decrypted images are as shown in Figure 10. It can be found that even if the original image is not successfully decrypted, the histograms of the decrypted image are still evenly distributed and can effectively resist attacks.

The second method is to use the indicator NBCR (number of bit change rate) to quantitatively illustrate the key sensitivity from another perspective [32]. The NBCR of matrices $B_1$ and $B_2$ is defined as formula (9):

$$\text{NBCR}(B_1, B_2) = \frac{\text{Ham}(B_1, B_2)}{\text{len}},$$

where $\text{Ham}(B_1, B_2)$ is the Hamming distance of the two matrices $B_1$ and $B_2$ and len is the total number of bits of $B_1$ and $B_2$.

If the NBCR result is close to 50%, it means that $B_1$ and $B_2$ are completely different matrices without any correlation. Specifically, the seven parameters $(\mu, x_0, x_{01}, x_{02}, X(0), Y(0), Z(0))$ in the algorithm are used as keys to form an infinite dimensional key space, and $K$ is slightly changed $(10^{-11})$ to obtain another key $K'$. Every parameter change in $(\mu, x_0, x_{01}, x_{02}, X(0), Y(0), Z(0))$ is in a very small $\delta$ interval. For example, the small change of $x_0$ is $x_0 + 10^{-11} \in [x_0 - \delta, x_0 + \delta] (\delta = 10^{-10})$, and other parameters remain unchanged. Encrypting the original image with the keys $K$ and $K'$, respectively, to obtain the encryption results $C_1$ and $C_2$, and then calculate the NBCR values of $C_1$ and $C_2$; the same encryption result is decrypted with keys $K$ and $K'$, respectively, to obtain decryption results $D_1$ and $D_2$, and then calculate the NBCR values of $D_1$ and $D_2$. The NBCR test results of the algorithm in this paper are shown in Table 10.

It can be seen from Table 10 that the calculated NBCR is very close to 50%, indicating that the results are completely different, which means that the algorithm is very sensitive to slightly changed keys.

4.2.5. Plaintext Sensitivity Analysis. The plaintext sensitivity analysis aims to analyze the degree of change of the ciphertext image caused by the slight change of the plaintext image under the same key condition. When the change is large, the encryption system has a strong plaintext sensitivity and can effectively resist differential attacks. NPCR (number of pixel change rate) and UACI (pixel average change intensity) are usually used to detect the ability of image encryption schemes to resist differential attacks. The calculation formulas (10) and (11) are as follows:

$$\text{NPCR} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} C(m, n)}{M \times N} \times 100\%,$$

$$\text{UACI} = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} |I_1(m, n) - I_2(m, n)|}{M \times N \times 255} \times 100\%.$$  

In the formula, $M \times N$ is the image size. Assuming that two plaintext images are different in only one pixel, after using the same algorithm to encrypt, the pixel values of the positions $(m, n)$ in the ciphertext image are, respectively, $I_1(m, n)$ and $I_2(m, n)$; when the value of the two is the same, then the value of $C(m, n)$ is 0, otherwise the value is 1. In the Lena plaintext, a pixel is randomly selected and its value is increased by 1, the same algorithm is used to encrypt, and the NPCR and UACI values are calculated. The results are shown in Table 11.

It can be seen from Table 11 that the NPCR of the image encrypted by the algorithm proposed in this paper is close to the ideal value of 100%, and NPCR is close to the ideal value of 33%. Compared with literature [33] and literature [34], the algorithm in this paper has stronger anti-differential attack capability.

Although the calculated NPCR and UACI values are close to the ideal values, only the NPCR and UACI values of the encrypted image are given quantitatively. In order to qualitatively analyze whether the encryption algorithm can

| Table 8: Comparison of global entropy of each channel before and after image encryption. |
|-----------------------------------------------|-----------------|-----------------|-----------------|
| R channel | G channel | B channel |
| Original image | 7.2682 | 7.5901 | 6.9951 |
| Ciphertext image | 7.9994 | 7.9993 | 7.9993 |
| Literature [29] | 7.9971 | 7.9974 | 7.9973 |
| Literature [30] | 7.9914 | 7.9915 | 7.9916 |

| Table 9: Local information entropy test. |
|-----------------------------------------------|-----------------|-----------------|
| Ciphertext image R channel | 7.9023 | Pass |
| Ciphertext image G channel | 7.9019 | Pass |
| Ciphertext image B channel | 7.9027 | Pass |

![Figure 9: Decrypted image when the key is wrong.](image-url)
produce ciphertext images that are secure enough and can resist differential attacks, Wu et al. proposed the method of randomness test for NPCR and UACI, analyzed their confidence intervals, and obtained critical value results under different parameter combinations. For detailed results, please refer to reference [35].

At the same time, two images of different sizes are taken. According to the abovementioned method, the NPCR and UACI values of the two ciphertext images are obtained and compared with the corresponding critical values \( \alpha = 0.05 \).

\[
\begin{array}{ccc}
\mu & 49.9967 & 50.0014 \\
x_0 & 50.0021 & 49.9956 \\
x_{01} & 50.0039 & 49.9996 \\
x_{02} & 49.9983 & 50.0044 \\
X(0) & 49.9989 & 49.9997 \\
Y(0) & 50.0012 & 50.0029 \\
Z(0) & 49.9961 & 50.0024 \\
\end{array}
\]

\textbf{Table 10:} Test results of NBCR (%).

The test results are shown in Table 12. From the data in the table, it can be seen that the NPCR value and the UACI value are both within the confidence interval, so it can be proved that the algorithm proposed in this paper is sufficient to resist differential attacks.

\textbf{Table 11:} Comparison of pixel change rate and average change intensity of each literature using the same image.

\[
\begin{array}{ccc}
\text{NPCR} (%) & \text{UACI} (%) \\
Paper algorithm & 99.61 & 33.44 \\
Literature [33] & 99.55 & 33.44 \\
Literature [34] & 99.60 & 33.40 \\
\end{array}
\]

\textbf{4.2.6. Key Space Analysis.} The algorithm proposed in this paper uses a total of three logistic chaotic mappings. Although the values of parameter \( \mu \) are the same, the initial values \( x_0 \) of each time are different. Therefore, \( \mu, x_0, \) and the three initial values \( X(0), Y(0), \) and \( Z(0) \) of the ML neuron chaotic system can all be used as the system keys. The image is decrypted by changing the number of digits after the decimal point of the key to get the key sensitivity \( 10^{-15}, 10^{-16}, 10^{-16}, 10^{-16}, 10^{-16}, 10^{-16} \) of the key \( \mu, x_0, x_{01}, x_{02}, X(0), Y(0), Z(0) \), respectively, so the key capacity \( 10^{15} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} = 10^{111} \) of the encryption algorithm is obtained. Literature [36] pointed out that the key space of a security cryptosystem should be greater than \( 2^{100} \) and both ENISA and NIST standards require a key size of at least \( 2^{112} \). The key space of the encryption scheme in this paper is larger than the above standards.
4.2.7. Robustness Analysis. This paper uses noise attack and cropping attack to analyze the robustness of the proposed image encryption algorithm. There is often a lot of noise in the transmission channel, which has a serious impact on the transmission of encrypted images, and even causes the decryption algorithm to fail to restore the identifiable original image. Therefore, an excellent encryption algorithm must have certain noise robustness. In the process of information transmission, the loss of encrypted image information data will occur from time to time, which is often referred to as cropping attack. Therefore, the image encryption algorithm needs to be able to resist the cropping

| Test image | Image size | Test content | Calculation results (%) | Confidence interval test |
|------------|------------|--------------|-------------------------|-------------------------|
| Cameraman  | 256 × 256  | NPCR         | 99.59                   | Pass                    |
| Cameraman  | 256 × 256  | UACI         | 33.35                   | Pass                    |
| Lena       | 512 × 512  | NPCR         | 99.61                   | Pass                    |
| Lena       | 512 × 512  | UACI         | 33.44                   | Pass                    |

Table 12: NPCR and UACI test (α = 0.05).

| Test image | Noise density | Paper algorithm | Literature [37] | Literature [38] |
|------------|---------------|-----------------|-----------------|-----------------|
| Lena       | 0.005         | 30.2474         | 31.4161         | 29.9684         |
| Lena       | 0.05          | 20.3605         | 21.5482         | 19.9800         |
| Lena       | 0.1           | 17.8897         | 18.5665         | 17.0283         |

Table 13: PSNR of the original image and decrypted image under noise attack.
attack, so that it can successfully decrypt and recover recognizable images. PSNR (peak signal-to-noise ratio) between the original image and the decrypted image is an important indicator to measure the quality of the decrypted image, and its definition is shown in formulas (12) and (13).

\[
\text{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i, j) - D(i, j))^2,
\]

(12)

\[
\text{PSNR} = 10 \times \log_{10} \frac{\text{MAX}_I^2}{\text{MSE}},
\]

(13)

where \( M \times N \) is the image size, \( P(i, j) \) and \( D(i, j) \) are the original image and decrypted image, respectively, and \( \text{MAX}_I^2 \) is the square of the maximum pixel value of the image.

In order to analyze the anti-noise capability of the encrypted image, salt and pepper noises with different noise densities were added to the encrypted image, and the PSNR of the original image and the decrypted image was calculated. The Lena encrypted images after adding noise densities of 0.005, 0.05, and 0.1 and the images decrypted using the proposed encryption algorithm are shown in Figure 11. It can be concluded from the figure that under the attack of salt and pepper noise, the decrypted image can still be recognized well. The PSNR value between the original image and the decrypted image under the noise attack is calculated, and the results are shown in Table 13.

![Figure 12: Decryption effect after the encrypted image part is cropped. (a) Crop 1/16 of the encrypted image; (b) crop 1/4 of the encrypted image; (c) crop 1/2 of the encrypted image; (d) decrypted image when cropped by 1/16; (e) decrypted image when cropped by 1/4; and (f) decrypted image when cropped by 1/2.](image)

**Table 14: PSNR of the original image and decrypted image under crop attack.**

| Test image | Crop | Paper algorithm | Literature [37] | Literature [38] |
|------------|------|-----------------|-----------------|-----------------|
| Lena       | 1/16 | 20.0312         | 20.0200         | 20.0820         |
|            | 1/4  | 14.2157         | 14.6140         | 14.3566         |
|            | 1/2  | 11.2968         | 11.8632         | 11.4038         |

**Table 15: Encryption speed comparison of images of different sizes (s).**

| Image size | Paper algorithm | Literature [39] | Literature [40] |
|------------|-----------------|-----------------|-----------------|
| 256 \( \times \) 256 | 0.4282          | 0.2736          | 0.4389          |
| 512 \( \times \) 512 | 1.7967          | 1.6166          | 1.8112          |
Experiments show that the proposed encryption algorithm has strong robustness against salt and pepper noise attacks. In order to analyze the anti-cropping attack capability of the encrypted image, the data of the encrypted image are crop by 1/16, 1/4 and 1/2, respectively, and the encryption algorithm proposed in this paper is used for decryption, respectively. The experimental effect is shown in Figure 12. As can be seen from the figure, under the crop attack, the decrypted image still has a certain degree of recognition. The PSNR value between the original image and the decrypted image under the crop attack is calculated, and the results are shown in Table 14. Experiments show that the proposed encryption algorithm has the ability to resist crop attack.

4.2.8. Speed Performance Analysis. In addition to considering security, encryption speed is also an important aspect to measure the pros and cons of encryption algorithms, especially for real-time network transmission requirements. For images of different sizes, the encryption speed of this algorithm and existing encryption algorithms is compared, as shown in Table 15. The computer used for testing is Intel(R) Core(TM)i5-8265U CPU (1.60 GHz), 8G memory. It can be seen from the table that the algorithm in this paper has certain advantages and high real-time performance, which is very necessary for the image encryption algorithm.

5. Conclusions
The encryption algorithm proposed in this paper combines an improved ML neuron model with DNA encoding. First, the original image and the logistic chaotic matrix are divided into blocks and DNA encoding is performed, respectively, and then the DNA operation and decoding between the same positions of the two image blocks is continued. Second, the continuous iteration of the chaotic sequence is used to perform determinant permutation on the matrix. Finally, the ciphertext image is obtained, in which the DNA encoding, operation, and decoding methods are determined by the chaotic sequence generated by the ML neuron chaotic system. Encryption performance analysis proves that the algorithm has a larger key space, better statistical characteristics, stronger key sensitivity and plaintext sensitivity, and can effectively resist various attacks such as exhaustive attacks, statistical analysis attacks, and differential attacks, indicating that it has good security and effectiveness.

Data Availability
The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (No. 61966022).

References
[1] Y. Q. Zhang, X. F. Wang, X. F. Liu, and L. Liu, “Survey on cloud computing security,” Journal of Software, vol. 27, no. 6, pp. 1328–1348, 2016.
[2] J. Xu, H. C. Wu, and X. F. Tao, “5G cyberspace security game,” Journal of Electronics and Information Technology, vol. 42, no. 10, pp. 2319–2329, 2020.
[3] C. M. Wu, “An improved discrete Arnold transform and its application in image scrambling and encryption,” Acta Physica Sinica, vol. 63, no. 9, pp. 91–110, 2014.
[4] X. Wang, X. Zhu, and Y. Zhang, “An image encryption algorithm based on Josephus traversing and mixed chaotic map,” IEEE Access, vol. 6, pp. 23733–23746, 2018.
[5] L. Teng, X. Wang, and J. Meng, “A chaotic color image encryption using integrated bit-level permutation,” Multimedia Tools and Applications, vol. 77, no. 6, pp. 6883–6896, 2017.
[6] M. Ghebleh and A. Kanso, “A novel efficient image encryption scheme based on chained skew tent maps,” Neural Computing & Applications, vol. 31, no. 7, pp. 2415–2430, 2019.
[7] M. Wang, X. Wang, Y. Zhang, and Z. Gao, “A novel chaotic encryption scheme-based on image segmentation and multiple diffusion models,” Optics & Laser Technology, vol. 108, pp. 558–573, 2018.
[8] L. Xu, X. Gou, Z. Li, and J. Li, “A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion,” Optics and Lasers in Engineering, vol. 91, pp. 41–52, 2017.
[9] X. X. Mao, K. H. Sun, and W. H. Liu, “Image encryption algorithm based on fractional order unified chaotic system,” Transducer and Microsystem Technologies, vol. 36, no. 6, pp. 138–141, 2017.
[10] J. D. Watson and F. H. C. Crick, “Molecular structure of nucleic acids: a structure for deoxyribonucleic acid,” Nature, vol. 171, no. 4356, pp. 737–738, 1953.
[11] M. A. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, “A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation,” Optics & Laser Technology, vol. 121, Article ID 105777, 2020.
[12] X. Xue, D. Zhou, and C. Zhou, “New insights into the existing image encryption algorithms based on DNA coding,” PLoS One, vol. 15, no. 10, Article ID e0241184, 2020.
[13] H. J. Tian, P. Lei, and Y. Wang, “Image encryption algorithm based on chaos and dynamic DNA coding,” Journal of Jilin University (Engineering and Technology Edition), vol. 44, no. 3, pp. 801–806, 2014.
[14] F. Shi, H. M. Zhang, and X. L. Zhang, “Image encryption algorithm based on chaotic map and DNA coding,” Computer Engineering and Applications, vol. 54, no. 5, pp. 91–95, 2018.
[15] X. Wei, L. Guo, Q. Zhang, J. Zhang, and S. Lian, “A novel color image encryption algorithm based on DNA Sequence operation and hyper-chaotic system,” Journal of Systems and Software, vol. 85, no. 2, pp. 290–299, 2012.
[16] Q. Zhang, L. Guo, and X. P. Wei, “Image encryption using DNA addition combining with chaotic maps,” Mathematical and Computer Modelling, vol. 52, no. 11-12, pp. 2028–2035, 2010.
[17] X. Su, W. Li, and H. Hu, “Cryptanalysis of a chaos-based image encryption scheme combining DNA coding and entropy,” Multimedia Tools and Applications, vol. 76, no. 12, pp. 14021–14033, 2017.
[18] X.-Y. Wang, P. Li, Y.-Q. Zhang, L.-Y. Liu, H. Zhang, and X. Wang, “A novel color image encryption scheme using...
DNA permutation based on the Lorenz system,” *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6243–6265, 2018.

[19] C. Morris and H. Lecar, “Voltage oscillations in the barnacle giant muscle fiber,” *Biophysical Journal*, vol. 35, no. 1, pp. 193–212, 1981.

[20] K. J. Wu, Y. Z. Shan, T. T. Du, and H. W. Lu, “Chaotic adaptive synchronization of coupled Morris-Lecar neurons and its application to chaotic secure communication,” *Journal of Lanzhou Jiaotong University*, vol. 35, no. 6, pp. 35–40, 2016.

[21] K. J. Wu and M. J. Yan, “Chaotic synchronization and application of neuron models with different structures,” *Study on Optical Communications*, vol. 6, pp. 23–27, 2021.

[22] Y. X. Wang, *The Dynamics Mechanism Research of Neuronal Excited Mode*, Hebei University of Technology, Tianjin, China, 2012.

[23] Y. Z. Shan, *Synchronization Research and Application of Coupled ML Neurons Model*, Lanzhou Jiaotong University, Lanzhou, Gansu, 2016.

[24] R. M. May, “Simple mathematical models with very complicated dynamics,” *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[25] G. Ye and X. Huang, “An efficient symmetric image encryption algorithm based on an intertwining logistic map,” *Neurocomputing*, vol. 251, pp. 45–53, 2017.

[26] D. Ravichandran, P. Praveenkumar, J. B. Balaguru Rayappan, and R. Amirharaj, “Chaos based crossover and mutation for securing DICOM image,” *Computers in Biology and Medicine*, vol. 72, pp. 170–184, 2016.

[27] X. Wu, K. Wang, and X. Wang, “Color image DNA encryption using NCA map-based CML and one-time keys,” *Signal Processing*, vol. 148, pp. 340–353, 2018.

[28] A. Yaghouti Niyat, M. H. Moattar, and M. Niazi Torshiz, “Color image encryption based on hybrid hyper-chaotic system and cellular automata,” *Optics and Lasers in Engineering*, vol. 90, pp. 225–237, 2017.

[29] P. Zhen, G. Zhao, L. Min, and X. Jin, “Chaos-based image encryption scheme combining DNA coding and entropy,” *Multimedia Tools and Applications*, vol. 75, no. 11, pp. 6303–6319, 2016.

[30] A. Kadir, M. Aili, and M. Sattar, “Color image encryption scheme using coupled hyper chaotic system with multiple impulse injections,” *Optik*, vol. 129, pp. 231–238, 2017.

[31] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, “Local Shannon entropy measure with statistical tests for image randomness,” *Information Sciences*, vol. 222, pp. 323–342, 2013.

[32] Z. Hua, Y. Zhou, and H. Huang, “Cosine-transform-based chaotic system for image encryption,” *Information Sciences*, vol. 480, pp. 403–419, 2019.

[33] Y. T. Hou, *Image Encryption Schemes Based on Chaos and DNA Encoding*, Dalian University of Technology, Liaoning, Dalian, 2018.

[34] H. Lan and R. S. Ye, “A novel image encryption algorithm based on chaos and DNA coding,” *Journal of Shantou University (Natural Science Edition)*, vol. 34, no. 4, pp. 13–35, 2019.

[35] Y. Wu, J. P. Noonan, and S. Agaian, “NPCR and UACI randomness tests for image encryption,” *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, vol. 1, no. 2, pp. 31–38, 2011.

[36] S. D. Li, C. Y. Wu, D. S. Wang, and Y. Q. Dai, “Secure multiparty computation of solid geometric problems and their applications,” *Information Sciences*, vol. 282, pp. 401–413, 2014.

[37] X. Wang and C. Liu, “A novel and effective image encryption algorithm based on chaos and DNA encoding,” *Multimedia Tools and Applications*, vol. 76, no. 5, pp. 6229–6245, 2017.

[38] K. Zhan, D. Wei, J. Shi, and J. Yu, “Cross-utilizing hyper-chaotic and DNA sequences for image encryption,” *Journal of Electronic Imaging*, vol. 26, no. 1, Article ID 013021, 2017.

[39] Z. Hua, S. Yi, and Y. Zhou, “Medical image encryption using high-speed scrambling and pixel adaptive diffusion,” *Signal Processing*, vol. 144, pp. 134–144, 2018.

[40] S. Cai, L. Huang, X. Chen, and X. Xiong, “A symmetric plaintext-related color image encryption system based on bit permutation,” *Entropy*, vol. 20, no. 4, p. 282, 2018.