Computer Science and Information Systems 21(4):1673–1697 https://doi.org/10.2298/CSIS240314053W

GAN-DNADE: Image Encryption Algorithm Based on Generative Adversarial Network and DNA Dynamic Encoding

Xi Wang

School of Artificial Intelligence and Software Engineering,
Nanyang Normal University, Nanyang 473061, China
352720214@qq.com

Abstract. Aiming at the problems such as small key space and incomplete color channel encryption in traditional image encryption, this paper proposes a novel image encryption algorithm based on generative adversarial network (GAN) and DNA dynamic encoding. This paper introduces GAN into random key generation, and uses GAN to learn and train the random key generated by hyperchaotic system. A parallel chaotic system is used to generate two sets of pseudo-random sequences, and DNA dynamic encoding is introduced to further transform them to generate a new sequence. The pixel-level diffusion and scrambling of images within and between channels are carried out by using random sequences. The experimental results show that the randomness of GAN can significantly expand the key space, and the proposed algorithm has significant advantages in the security and anti-attack ability of ciphertext images.

Keywords: image encryption, generative adversarial network, DNA dynamic encoding, hyperchaotic system.

1. Introduction

Image has been widely used in Internet because of its characteristics of large amount of information and convenient transmission. Due to the characteristics of parallel information processing, optical information system has shown incomparable advantages of digital information system in mass information processing, especially the more complex the image processing and the more information, the more obvious this advantage. Night vision technology uses the two conditions of low light and infrared light to convert the low light or infrared light information invisible to the human eye from the target into visible light that can be felt by the human eye through signal sensing, acquisition, processing and display technology [1,2]. Among them, low-light level TV and low-light night vision play an important role in the monitoring of the enemy's fixed targets and the warning and security of our important targets [3,4]. However, because the low-light level image contains key information such as public security, military and space technology, the importance of protecting low-light level image from malicious attacks has gradually become prominent, and the encryption technology of low-light level image has been widely concerned by the international academic circle. In addition, unlike the visible image, the low-light image has complex noise, fuzzy texture characteristics, uneven illumination, low signal-to-noise ratio and less gray level. The traditional visible image encryption technology directly uses

the random sequence generated by chaotic system to encrypt, which has shortcomings in security. With the deepening of the research, the security risks brought by the linear relationship of optical transformation are gradually exposed [5,6].

Chaotic systems are highly sensitive to initial values and control parameters, with ergodicity, pseudo-randomness and unpredictability, so they have been widely used in image encryption [7,8]. The nonlinear characteristics of chaotic systems can effectively offset the security risks caused by linear transformations in optical encryption. However, some chaotic based encryption schemes often have some shortcomings [9], such as short cycle length caused by the limited precision of computers, which is one of the important problems of chaotic key stream generators [10]. In order to solve the randomness and security of keys, researchers have proposed many key generation schemes for chaotic systems or various other technologies [11,12].

With the rapid development of artificial intelligence technology, deep learning has been widely used in various fields of information security. Image information security in the era of artificial intelligence has attracted more and more attention from researchers [13]. Since generative adversarial networks (GANs) were first proposed in 2014 [14], GANs have become a hot topic in computer vision [15], natural language processing, and malicious attack detection [16]. Due to the randomness and difference of GAN training results, this paper uses chaotic sequences as training sets and control generators to obtain data samples with more randomness than chaotic sequences, so as to avoid problems such as long iteration period and time consuming of high-dimensional chaotic systems. It can obtain more random data samples than chaotic sequences, and speed up the generation of key, which saves time for batch encryption of low-light level images, and can meet the needs of real-time processing of low-light level images containing important military secrets.

Chaotic systems began to be applied to image encryption in the 1990s. Since then, image encryption based on chaotic system has been paid more and more attention. In recent years, researchers are still committed to improving the rationality, practicability and security of chaotic image encryption. Sekar et al. [17] designed an image encryption algorithm based on Deoxyribo Nucleic Acid (DNA) sequence manipulation and space-time chaos. The plaintext image was first converted into a DNA matrix and then scrambled. After multiple DNA sequence operations, the resulting matrix was finally converted into a ciphertext image. In 2019, using discrete chaotic mapping, Wang et al. [18] proposed an image encryption algorithm based on DNA sequence manipulation. The algorithm scrambled DNA horizontally and generated ciphertext images by XOR of scrambles matrix. In 2023, Li et al. [19] designed an image encryption algorithm using DNA computing, chaotic system and hash algorithm, which carried out DNA-level scrambling and diffusion of plaintext images. With the development of image encryption technology, there are also researchers working on related cryptanalysis work. For scramble-only image encryption algorithms, Wang et al. [20] proved that such encryption algorithms were not secure. In 2018, for the hyperchaotic image encryption algorithm using DNA computing, Fang et al. [21] pointed out the rationality, practicability and security problems, and cracked it through a selective plaintext attack. For the relevant tests that were often used to evaluate the security of image encryption algorithms, Zhang et al. [22] confirmed that these tests were only necessary conditions to ensure the security of the algorithm, but not sufficient conditions. For the image encryption algorithm based on chaotic mapping

and DNA coding, Yang et al. [23] simplified it into a substitution-scrambling structure and used selective plaintext attacks to crack it. In addition, it was particularly noteworthy that Singh et al. [24] reviewed some representative works in the field of image encryption and its cryptanalysis, and classified and summarized these works. More importantly, they also pointed out some challenging problems in the field of image encryption and its cryptanalysis.

There is no doubt that the problems identified in cryptanalysis work will be taken seriously by researchers when designing new image encryption algorithms. Therefore, the cryptanalysis of image encryption algorithm can promote the development and perfection of image encryption technology. So a novel image encryption algorithm based on generative adversarial network (GAN) and DNA dynamic encoding is proposed in this paper.

2. Related Works

In view of the low efficiency and poor security of traditional encryption algorithms, researchers have proposed many efficient image encryption algorithms based on the randomness of low-dimensional chaotic systems [25]. However, the structure of low-dimensional chaotic systems is too simple, which makes the generated pseudo-random sequences vulnerable to attacks. Yu et al. [26] proposed an image encryption algorithm based on the hyperchaotic system. The pseudo-random sequence generated by the hyperchaotic system had good randomness and wide range, which made the security of the encryption method significantly improved compared with the traditional low-dimensional chaotic system, but the image anti-attack ability was weak. Ouguissi et al. [27] introduced scrambling methods such as Arnold transform and bit combination scrambling respectively to improve security, but changing pixel positions did not destroy the statistical characteristics of the original image, resulting in low anti-statistical attack capability of the encrypted image. Wang et al. [28] put forward a synchronous scrambling diffusion algorithm based on bit-plane decomposition technology. This method had good encryption effect, but did not consider the problem of uneven distribution of bit-plane information. Wei et al. [29] proposed a selective encryption method based on the bit-plane information distribution. The algorithm had small computation, but had general security performance and low key space.

2.1. Randomness of Neural Networks

An artificial neural network consists of a large number of neurons connected to each other, each node represents a specific output function, and the connections between different nodes represent a weighted value of the signal passing through that connection. The output of the network is determined by the connection mode, weight value and excitation function of the network [30].

Suppose that the input $X = (x_1, x_2, \cdots, x_n)$ is a vector of length n, and the output $Y = (y_1, y_2, \cdots, y_n)$ is the result of training the neural network. V_e and W_s are the weights of the input layer and the output layer. V_{oe} and $W_{0,s}$ are input and output errors respectively. The forward training model of the neural network structure is:

$$y_i = g(w_{0,s} + \sum_{j=1}^{n} Z_j \cdot W_{s,j}). \tag{1}$$

Where $w_{0,s}$ is the offset of the output layer. $W_{s,j}$ represents the weight of the hidden layer node j to the output layer. The calculation formula from the hidden layer to the output layer is as follows:

$$Z_j = F(V_{oe,j} + x_k \cdot V_{e,j}). \tag{2}$$

Where $1 \leq j \leq n$, $1 \leq k \leq n$. Z_j represents the $j-th$ hidden layer node. $V_{oe,j}$ represents the bias of the $j-th$ node. $V_{e,j}$ represents the weight of the $j-th$ node. In order to make the network more powerful and generate a nonlinear mapping from input to output, the activation function selects the nonlinear function $F(\theta) = tanh(0\theta)$ so that it can learn complex data. The training error is:

$$er_i = y_i - x_i. \tag{3}$$

The backpropagation gradient process is:

$$\begin{cases} \sigma_{0,i} = a \cdot er_i \quad i = 1, 2, \cdots, n \\ \sigma_{h,j} = Z_j(1 - Z_j) \sum_{i=1}^{n} \sigma_{0,i} W_{s,j} \quad k = 1, 2, \cdots, n \end{cases} \tag{4}$$

The training process iterates by updating each weight and bias by backpropagating the gradient. Since there are many different neuron connections in a neural network, the overall behavior of the network system depends on the characteristics of each neuron and the interactions of different neurons. When the internal parameters of the neural network are randomly initialized, the stochastic gradient descent algorithm ensures the training randomization of different objects. Moreover, with the further complexity of the depth and structure of the neural network, the complexity of the internal parameter combination increases exponentially, which makes the parameter change of each neuron affect the output of the whole network. According to this characteristic, the randomness of neural network is used to further expand the random selection space of key and improve the security of encryption algorithm.

2.2. The Generation of Chaotic Sequences

For the plaintext image P with size $M \times N$, its hash value K is calculated by Keccak algorithm [31] and equally divided into 32 bytes, that is, $k_1, k_2, \cdots, k_{32}$. Using the following formula:

$$h_i = ((k_{j+1} \oplus k_{j+2} \oplus k_{j+3}) + k_{j+4} + k_{j+5} + k_{j+6})/256. \tag{5}$$

$$\begin{cases} x_0 = x_0' + 1 + abs(round(h_1) - h_1) \\ y_0 = y_0' + 1 + abs(round(h_2) - h_2) \\ z_0 = z_0' + 1 + abs(round(h_3) - h_3) \end{cases} \tag{6}$$

The initial state values x_0, y_0, z_0 of the chaotic system are calculated. Where, $i = 1, 2, 3$, $y = 6(i-1)$, $\oplus$ is the bitwise XOR operation. x_0', y_0', z_0' are the given value., $abs(r)$ is the absolute value of the parameter r, and $round(r)$ is the rounding of the parameter r. Input x_0, y_0, z_0 into the Lorenz chaotic system to generate chaotic sequences x, y, z. The used Lorenz chaos system is:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = -xz + bx - y \\ \dot{z} = xy - cz \end{cases} \tag{7}$$

System control parameter is $(a, b, c) = (10, 28, 8/3)$. Finally, we adopt,

$$\begin{cases} X = mod(floor(10^{10} \times (x - floor(x))), N) + 1 \\ Y = mod(floor(10^{10} \times (y - floor(y))), M) + 1 \\ Z = mod(floor(10^{10} \times (z - floor(z))), 256) \end{cases} \tag{8}$$

Convert x, y, z to the sequence X, Y, Z. Where $mod(r_1, r_2)$ is to perform modular r_2 operation on the parameter r_1, and $floor(r)$ is to round down the parameter r.

2.3. GAN Key Generation Method

The randomness of key has become an important factor affecting the security of cryptosystem. GAN is a class of artificial intelligence algorithms for unsupervised machine learning, which consists of two neural networks that compete with each other in a zero-sum game framework [32].

$$\min_G \max_D v(G, D) = E_{x \sim p_d}(x)[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \tag{9}$$

Where G is the generating network. D is the discrimination network. $v(G, D)$ is the loss function. x is the training set. $E_{x \sim p_d}$ is the distribution of the actual input data. $\log D(x)$ is the judgment value of the discriminator. $D(x)$ is the D network model used to determine whether the actual data and training z are noise from the input G network model. $E_{z \sim p_z(z)}$ is the distribution of noise data. $\log(1 - D(G(z))$ is the judgment value of the generated data. $G(z)$ is the data generated for the G network model. Through continuous game of maximum and minimum values, G network model and D network model are optimized alternately until the two models reach Nash equilibrium.

Quantum dots and quantum cellular automata are novel nanoelectronic devices that transmit information through coulomb interactions [33]. Compared with traditional technology, quantum cellular automata has the advantages of ultra-high integration, ultra-low power consumption and leadless integration. In recent years, scholars at home and abroad have used the structure of cellular neural networks and quantum cellular automata to construct QCNN on the basis of Schrodinger equation. Due to the quantum interaction between quantum dots, QCNN can obtain complex linear dynamic properties from the polarizability and quantum phase of each quantum cellular automaton, which can be used to construct nanoscale hyperchaotic oscillators. For a QCNN coupling two elements, it can be described by the following differential equation:

$$\dot{g}_1 = -2a_1\sqrt{1-g_1^2}\sin h_1. \tag{10}$$

$$\dot{h}_1 = (-b_1(g_1-g_2) + 2a_1 g_1 \cos h_1)/\sqrt{1-g_1^2}. \tag{11}$$

$$\dot{g}_2 = -2a_2\sqrt{1-g_2^2}\sin h_2. \tag{12}$$

$$\dot{h}_2 = (-b_2(g_2-g_1) + 2a_2 g_2 \cos h_2)/\sqrt{1-g_2^2}. \tag{13}$$

In the formula, g_1, g_2 are the polarizability. h_1, h_2 are the quantum phase. a_1, a_2 are the proportional coefficients of the energy between the midpoints of each unit. b_1, b_2 are the weighted influencing factors of the differences in the polarizability of adjacent units. When $a_1 = a_2 = 0.28$, $b_1 = 0.7$, $b_2 = 0.3$, the system is in a chaotic state.

Because QCNN is a high order hyperchaotic system, the iteration speed is slow and the computation is large. The hyperchaotic random matrix generated by QCNN is input into GAN as the training set, and the random data is learned to be used as the secure GAN key pool of the encryption system. The random numbers generated after training have similar characteristics to the random numbers generated by the chaotic system, that is, the new random numbers also have chaotic characteristics. The sensitivity is that when different chaotic random sequences generated by QCNN with different initial values or control parameters are used as learning objects, GANs will also learn and generate completely different random numbers. However, the random numbers generated by GANs also have some characteristics that are different from those of chaotic systems. For example, because the learning and training process is unsupervised, the generated random sequence of numbers is not subject to human control, that is, it has non-repeatability. In short, GAN key generation method can effectively improve the speed of key generation on the premise of guaranteeing the key characteristics.

3. Proposed Image Encryption Method

The GAN key generation method is applied to a new low-light level image encryption algorithm, and a 2-D coordinate pointer calculation method is designed. In each encryption, a specific location is calculated from the key pool using the plaintext related control parameters and user-set parameters, and then different random sequences are obtained from the key as scrambled keys. Only one scrambling encryption framework is given in the description of the algorithm in this paper, but in practice each scrambling operation is to set different operations related to plaintext and users, and the attacker cannot break the encryption algorithm through known plaintext or select plaintext attacks.

The key generated by chaotic system is trained by GAN to obtain GAN key pool, and two chaotic random phase masks are selected from GAN key pool for diffusion stage. The overall encryption process is shown in Figure 1. RM_1 and RM_2 are two phase masks. The original image is scrambled first, and then double random phase fractional Fourier transform (FFT) [34] is performed to obtain the final ciphertext image.

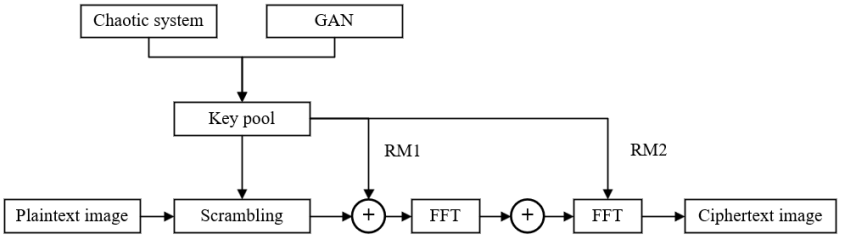


Fig. 1. Proposed encryption process

3.1. Pixel Replacement Based on DNA Dynamic Encoding

DNA dynamic encoding is performed on scrambled image T. Each pixel is encoded into 4 bases using different coding rules to obtain the DNA encoding sequence E. The encoding rule $R_{i,j}$ for each pixel depends on the location (i, j) of the pixel $T_{i,j}$ and the sequence Z.

$$R_{i,j} = (mod((i-1) \times N + j, 8) \oplus mod(Z_{(i-1)\times N+j}, 8)) + 1. \tag{14}$$

Where $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$. The DNA sequence was downloaded from the GenBank database and $4 \times M \times N$ bases are intercepted. DNA XOR operations are performed on these bases with the DNA encoding sequence E. Finally, encoding rule is used to decode the DNA of the operation result and reassemble it into an intermediate ciphertext image I.

3.2. Pixel Row Diffusion

It reorganizes the sequence Z into an $M \times N$ matrix in a column-first manner. Row spread operations are carried out in the direction of rows in the form of column vectors.

$$\begin{cases} C'_1 = I_1 \oplus I_N \oplus I_{N-1} \oplus Z_1 \\ C'_2 = I_2 \oplus C'_1 \oplus I_N \oplus Z_2 \\ C'_i = I_i \oplus C'_{i-1} \oplus C'_{i-2} \oplus Z_i \end{cases} \tag{15}$$

Where, C'_i is the $i-th$ column of the middle ciphertext image C' obtained after the row diffusion operation. I_i and Z_i are columns I and Z, respectively, $i = 3, 4, \cdots, N$. Then the column diffusion operation is carried out to obtain the final ciphertext image C.

$$\begin{cases} C_1 = C'_1 \oplus C'_M \oplus C'_{M-1} \oplus Z_1 \\ C_2 = C'_2 \oplus C_1 \oplus C'_M \oplus Z_2 \\ C_i = C'_i \oplus C_{i-1} \oplus C_{i-2} \oplus Z_i \end{cases} \tag{16}$$

Where C_i is row $i-th$ of C obtained after the column diffusion operation. C'_i and Z_i are row $i-th$ of C' and Z respectively, $i = 3, 4, \cdots, M$.

Case 1. When the chaotic sequence x is converted to sequence X, the modulus used by the original formula (6) is 256, not the column number N of the image.

Analysis. X in matrix form is used to implement line-by-line scrambling of pixels. Therefore, the Josef traversal variable step size should have a value range of $[1, N]$, not $[1, 256]$. Otherwise, when $N \gg 256$, the scrambling of pixels will be limited to a small range. The same is true for the transformation of chaotic sequence y.

Case 2. The DNA encoding rule R_{ij} is calculated according to the original formula (14), and its value range is $[0, 7]$. In addition, the chaotic sequence element used in the original formula (14) is $Z_{(i-1)\times N+1}$.

Analysis. According to **Algorithm 1**, the value range of the coding rule should be $[1, 8]$. In addition, in order to maximize the randomness and dynamics of the encoding rules, Z should be used more fully. Therefore, R_{ij} should be calculated using $Z_{(i-1)\times N+j}$ in equation (14).

Algorithm 1 Pixel diffusion effect elimination algorithm

Input: the ciphertext image C with size $M \times N$.
step 1: When $i \in [3, M]$, it repeats operation $C^1(i,:) = C(i,:) \oplus C(i-1,:) \oplus C(i-2,:)$.
step 2: Execute operation $C^1(2,:) = C(2,:) \oplus C(1,:) \oplus C^1(M,:)$.
step 3: Execute operation $C^1(1,:) = C(1,:) \oplus C^1(M,:) \oplus C^1(M-1,:)$.
step 4: When $j \in [3, N]$, it repeats operation $C^2(:,j) = C^1(:,j) \oplus C^1(:,j-1) \oplus C^1(:,j-2)$.
step 5: Execute operation $C^2(:,2) = C^1(:,2) \oplus C^1(:,1) \oplus C^2(:,N)$.
step 6: Execute operation $C^2(:,1) = C^1(:,1) \oplus C^2(:,N) \oplus C^2(:,N-1)$.

Case 3. The description of the diffusion process is inconsistent. In addition, if the number of rows or columns in the input image is less than 4, the diffusion process will not work properly.

Analysis. According to the display of row scrambling, it can be seen from the formula (15) that the row diffusion is carried out in the form of column vectors. That is, at the same time, the pixels on a column are correspondingly spread to other columns, which is to spread the column in the direction of the row. When the formula (15) is explained in the original paper, P_i also represents the i row of the image matrix. Similarly, columns can be diffused using equation (15).

The problem of the diffusion process not working properly is discussed only $N < 4$; $M < 4$ is similar. When $N = 1$, P_{N-1} in equation (15) is meaningless. When $N = 2$, equation (15) degenerates to:

$$P_i' = \begin{cases} (P_2 \oplus Q_1) mod 256, i = 1 \\ P_1 mod 256, i = 2 \end{cases} \tag{17}$$

Obviously, this is also the result of unreasonable and meaningless diffusion. When $N = 3$, equation (15) degenerates into:

$$P'_i = \begin{cases} (P_1 \oplus P_3 \oplus P_2 \oplus Q_1) mod 256, i = 1 \\ (P_2 \oplus P'_1 \oplus P_1 \oplus Q_1) mod 256, i = 2 \\ (P_2 \oplus P'_2 \oplus P'_1 \oplus Q_3) mod 256, i = 3 \end{cases} \tag{18}$$

According to equation (18), diffusion is irreversible. So if it knows P' and Q, it can only find P_3, it can not find P_1 and P_2. In addition, the module 256 operation in equation (15) is also redundant. Q_2 should also be used when diffusing P_2.

Case 4. GAN-DNADE downloads DNA sequences from the GenBank database and intercepts $4 \times M \times N$ bases.

Analysis. To encrypt a plaintext image with a size of $M \times N$, the encryptor must obtain at least $4 \times M \times N$ bases securely from the GenBank database. The same goes for the decryption party to complete the decryption. In other words, a plaintext image with a size of $M \times N$ is transmitted through an insecure channel, and GAN-DNADE is selected to achieve the encryption protection of the image, and both the encryptor and the decryptor need to securely download at least the same length of data from a third party. Obviously, this design makes encryption pointless. A reasonable design is to use a chaotic system to generate the required base data.

Case 5. By simple processing of ciphertext images, the encryption structure can be degraded from scramble-replace-diffusion structure to scramble-replace structure.

Analysis. As can be seen from equations (14) and (15), the diffusion process of GAN-DNADE not only has the effect of pixel diffusion, but also has the effect of pixel replacement caused by the difference with chaotic matrix Z. Since the ciphertext image C is known, $C'_i \oplus Z_i$ can be obtained according to equation (16).

$$C'_i \oplus Z_i = C_i \oplus C_{i-1} \oplus C_{i-2}. \tag{19}$$

Where $i = 3, 4, \cdots, M$. The same can be done for row 1 and row 2 as in equation (20).

$$\begin{cases} C'_1 \oplus Z_N \oplus Z_{N-1} \oplus Z_1 = C_1 \oplus (C'_N \oplus Z_N) \oplus (C'_{N-1} \oplus Z_{N-1}) \\ (C'_2 \oplus Z_2 \oplus Z_N = C_2 \oplus C_1 \oplus (C'_N \oplus Z_N) \end{cases} \tag{20}$$

At this time, although C' can not be directly obtained, the pixel diffusion effect of the column diffusion process can be completely eliminated, so that the processed ciphertext only has the replacement effect generated by the XOR with Z. Similarly, the pixel diffusion effect of the line diffusion process can also be eliminated, and finally a ciphertext image with only pixel replacement effect can be obtained. At this point, GAN-DNADE has degenerated into a scramble-replace replacement structure. However, the encryption effect of continuous secondary replacement is no different from that of a single replacement.

3.3. Multi-scale Feature Extraction and Fusion Module

In order to obtain more context information based on the existing image data and improve the feature extraction effect in the image, according to the idea of ASPP (Atrous Spatial

Pyramid Pooling) module, multi-scale features are extracted in parallel by using cavity convolution with different expansion rates, and the extracted multi-scale feature maps are fused. The specific structure of the multi-scale feature extraction and fusion module used by the network in this paper is shown in Figure 2. In figure 2, k is the convolution kernel size, s is the step size, p is the filling range, and d is the expansion rate. The features are extracted by using cavity convolution with expansion rates of 6, 12, and 18 respectively. In order to prevent the convolution degradation caused by excessive void rate, it is fused with the features after global average pooling. The use of this module increases the receptive field and obtains more context information, which helps to improve the feature extraction ability of the model for boundary details.

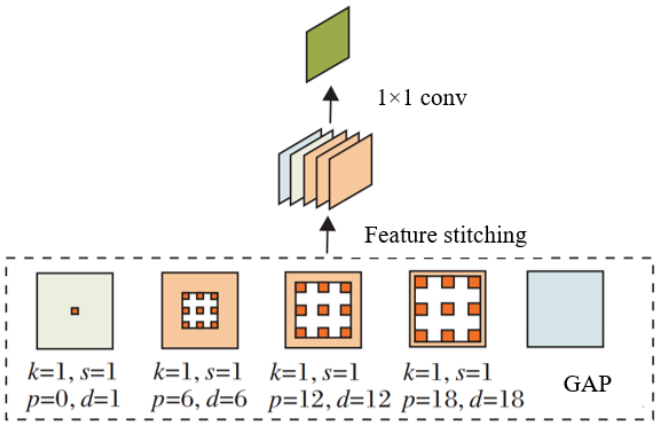


Fig. 2. Multi-scale feature extraction fusion diagram

In this paper, the features of different scales extracted by different expansion rates are spliced, and then 1×1 convolution is used to transform them into fixed-size feature maps. Because of the fusion of features of different scales, the ability of the network to extract features in the image is improved, and the loss of feature information is reduced, which is conducive to the subsequent sampling and recovery of detail information on the decoder.

3.4. Attention Module

At present, image semantic segmentation networks based on deep learning usually adopt multi-scale feature fusion or U-Net structure fusion of low-level and high-level semantic features, without considering the relation and correlation between each position or channel of feature map. Therefore, the network in this paper respectively conducts correlation modeling for the location dimension and channel dimension of the feature map. Firstly, Non-local [35] structure is used to achieve correlation modeling for the location of the feature space. Then a Non-local structure is used to model the correlation between channels, and the output results of the two modules are added and fused to obtain the global dependency relationship between features, and improve the ability of the network

to distinguish between normal organization areas, abnormal organization areas and backgrounds. It improves the segmentation effect of the transition region between normal and abnormal organization, and alleviates the adverse effects of void multiple on network learning to a certain extent.

In order to make use of the correlation between features of different regions and enhance each other's expression of their features, the network in this paper uses a modeling method similar to that in Non-local to model the location, as shown in Figure 3. Firstly, the correlation strength matrix between the features of any two points is calculated, that is, the original feature A is convolved to obtain the feature B (dimension is $(H \times W) \times C'$) and the feature C (dimension is $C' \times (H \times W)$), and the matrix product is carried out to obtain the correlation strength matrix between the features of any two points (dimension is $(H \times W) \times (H \times W)$). After softmax normalization operation, the attention diagram S of each position for other positions is obtained. The more similar the two features, the greater the response value. The response value in the location attention map is used as the weight to fuse the feature D, and the feature can be selectively enhanced or suppressed by the correlation between pixels.

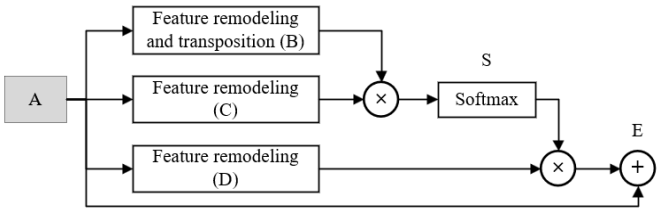


Fig. 3. Location attention module block diagram

For feature graph $S \in R^{N \times N}$,

$$S_{ij} = \frac{exp(B_i C_j)}{\sum_{i=1}^{N} exp(B_i C_j)}. \tag{21}$$

Where, C is the number of characteristic channels. H and W are the height and width of the feature graph. $N = H \times W$. S is the influence of the $i - th$ position on the $j - th$ position, that is, the degree of correlation/correlation between the $i - th$ position and the $j - th$ position, the larger the more similar.

Location attention module output feature map $E \in R^{C \times H \times W}$:

$$E_j = \alpha \sum_{i=1}^{N} (S_{ij} D_i) + A_j. \tag{22}$$

Where α is the scale coefficient, initialized to 0, and updated weights are assigned by gradual learning. E is the weighted sum of all location feature similarity and original location, with global context information.

The feature maps between different channels can be regarded as responses to specific categories, and there are certain dependencies between different responses. In order to

model this dependency explicitly, inspired by the Non-local module, the network in this paper conducts similar attention modeling for the channel dimension, as shown in Figure 4.

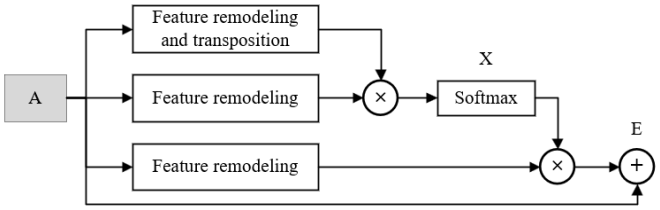


Fig. 4. Channel attention module diagram

By modeling the relationship between channels, the specific semantic response capability under channels is enhanced. The specific process is similar to location attention modeling, but the difference is that when obtaining the feature attention diagram $X \in R^{C\times C}$, dimensionality transformation and matrix product of any two channel features are performed to obtain the correlation strength of any two channels. The attention diagram between channels is also obtained through softmax operation. The attention force weighting between channels is used for fusion, so that each channel has a global association, and the features of stronger semantic response are obtained. The specific calculation process is as follows:

$$x_{ij} = \frac{exp(A_i A_j)}{\sum_{i=1}^{C} exp(A_i A_j)}. \tag{23}$$

$$E_j = \beta \sum_{i=1}^{C} (x_{ij} A_i) + A_j. \tag{24}$$

Where C is the number of characteristic channels. H and W are the height and width of the feature graph. x_{ij} is the influence of channel i on channel j. β is the scale coefficient, initialized to 0, updated by iterative learning, and the final output E is the weighted sum of each channel feature and the original feature. In this paper, we strengthen the sensitivity of the network to the boundary by means of intra-class feature response and interclass feature suppression.

3.5. Cryptographic Analysis

Based on the above analysis, GAN-DNADE can be described as:

$$C = f_3(f_2(f_1(P, K), K), K). \tag{25}$$

Where $f_1(r_1, r_2)$ represents the scrambling operation of input image r_1 under the control of secret key r_2. $f_2(r_1, r_2)$ indicates the replacement of r_1 under the control of r_2. $f_3(r_1, r_2)$ indicates the diffusion operation performed on r_1 under the control of r_2.

A simple processing of C can deform GAN-DNADE into a scramble-replace structure. Therefore, equation (25) can be further simplified as:

$$C = f_2'(f_1(P, K), K). \tag{26}$$

$f_2'(r_1, r_2)$ indicates the replacement of r_1 under the control of r_2. The following only discusses GAN-DNADE with the pixel diffusion effect eliminated. The scrambling procedure has no encryption effect on plaintext images with a single pixel value. For a plaintext image with a single pixel value, equation (26) can be simplified as:

$$O = f_2'(I, K). \tag{27}$$

Where I represents the input plaintext image of a single pixel value. O indicates the output ciphertext image. Therefore, $f_2'(r_1, r_2)$ can be determined using the plaintext image with a single pixel value and its corresponding ciphertext image, that is, the equivalent replacement matrix E^S can be determined. Taking a plaintext image of size 2×2 as an example, it uses the following formula:

$$P^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{28}$$

and corresponding ciphertext image:

$$C^0 = \begin{bmatrix} c_{0,1} & c_{0,2} \\ c_{0,3} & c_{0,4} \end{bmatrix} \tag{29}$$

It can determine the result of the replacement of 0 value pixel at each position. That is, the 0 value pixel is replaced by $c_{0,1}$ at $(1, 1)$, $c_{0,2}$ at $(1, 2)$, $c_{0,3}$ at $(2, 1)$, and $c_{0,4}$ at $(2, 2)$. Similarly, a plaintext image with a single pixel value of $v = 1, 2, \cdots, 255$ and its corresponding ciphertext image can be used to determine the replacement result of a pixel with a value of v at each position.

$$P^v = \begin{bmatrix} v & v \\ v & v \end{bmatrix} \tag{30}$$

$$C^v = \begin{bmatrix} c_{v,1} & c_{v,2} \\ c_{v,3} & c_{v,4} \end{bmatrix} \tag{31}$$

By stretching all C^v into one-dimensional row vectors and arranging them from top to bottom, an equivalent replacement matrix is obtained for any ciphertext image of size 2×2.

$$E^S = \begin{bmatrix} c_{0,1} & c_{0,2} & c_{0,3} & c_{0,4} \\ c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{255,1} & c_{255,2} & c_{255,3} & c_{255,4} \end{bmatrix} \tag{32}$$

Therefore, for any ciphertext image with a size of 2×2 and generated by K, the pixel replacement effect can be eliminated by E^S. Similarly, the above cryptanalysis procedure can be applied to a ciphertext image of size $M \times N$ to construct an equivalent replacement matrix of size $256 \times (M \times N)$.

$$E^S = \begin{bmatrix} c_{0,1} & c_{0,2} & \cdots & c_{0,M\times N-1} & c_{0,M\times N} \\ c_{0,1} & c_{1,2} & \cdots & c_{1,M\times N-1} & c_{1,M\times N} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{255,1} & c_{255,2} & \cdots & c_{255,M\times N-1} & c_{255,M\times N} \end{bmatrix} \tag{33}$$

In the first example of E^S, each pixel $c_i^*(i = 1, 2, \cdots, M \times N)$ of any ciphertext image is found, thus determining the value of each pixel before the replacement operation. In this way, the pixel replacement effect of the GAN-DNADE can be eliminated, further degrading it to only pixel scrambling effect. Scramble-only image encryption algorithm has been proved to be insecure. In this paper, a simpler method is chosen to obtain the equivalent scramble-matrix E^S. First, replace the first 255 pixels of an all-zero valued plaintext image of size $M \times N$ with $1, 2, \cdots, 255$. After obtaining the corresponding ciphertext image, the pixel diffusion and replacement effect of the ciphertext image are eliminated. The processed ciphertext image has only pixel scrambling effect, in which the corresponding position of each non-zero plaintext pixel can be found. In this way, it is possible to determine the position of the first 255 non-zero pixels of the plaintext image after scrambling. Similarly, the position of the remaining $M \times N - 255$ plaintext pixels can be determined after scrambling. The position of up to 255 pixels can be determined at a time, so a maximum of $floor(M \times N/255) + 1$ selected plaintext image and its corresponding ciphertext image are required to determine E^S.

As can be seen the above analysis, GAN-DNADE can be fully cracked using up to $256floor(M \times N/255) + 1$ selective plaintext images and their corresponding ciphertext images. The following is a specific selective plaintext attack algorithm.

Algorithm 2 Selective plaintext attack algorithm

Input: To restore the ciphertext image C of the plaintext image, the size of which is $M \times N$.
step 1: When $v \in [1, 256]$, it repeats operations:
(a) A plaintext image P^{v-1} with a single pixel value of $v-1$ is constructed, and its corresponding ciphertext image C^{v-1} is obtained by encryption.
(b) Calling the Algorithm 1 to eliminate the pixel diffusion effect of C^{v-1}, stretch C^{v-1} into a one-dimensional row vector, and let $E^S(v, :) = C^{v-1}$.
step 2: Determine the number of plaintext image q required to obtain the equivalent scrambling matrix E^P.
step 3: When $w \in [1, q]$, it repeats operations:
(a) Construct the $w - th$ selected plaintext image P^w required for E^P acquisition, and encrypt to obtain its corresponding ciphertext image C^w.
(b) Calling the Algorithm 1 to eliminate the effect of C^w pixel diffusion.
(c) Using E^S to eliminate the pixel replacement effect of C^w.
(d) Finding the corresponding position of each plaintext image non-zero value pixel in C^w and save it to E^P.
step 4: Algorithm 1 is invoked to eliminate the pixel diffusion effect of C and C^2 is obtained.
step 5: Using E^S to eliminate the pixel substitution effect of C^2, C^3 is obtained.
step 6: Using E^P to eliminate the pixel substitution effect of C^3, P^R is obtained.
Output: Recovering plaintext images P^R as well as E^S and E^P.

3.6. Image Decryption Process

The decryption process is the inverse process of the encryption process, which is mainly divided into three steps: pixel level inverse diffusion, reverse bit level scrambling diffusion, and pixel level inverse scrambling of the image. When the receiver receives the ciphertext image C', and the two sequences Z_1 and Z_2, the decryption process can be implemented by following steps:

Step 1. Decompose the secret image C' to the three-color channel, denoted as C_R, C_G and C_B, splice the color channel image and carry out pixel-level inverse diffusion to obtain the pixel matrix P. The pixel matrix is decomposed into 8 bit planes, and the bits between bit planes are diffused in reverse bit level.

Step 2. The vertical reverse diffuses bits of the bit planes 2, 4, 6, 8, while the lateral reverse diffuses bits of the bit planes 1, 3, 5, 7, converting bit-level bits to the pixel level.

Step 3. Channel image R is transformed by inverse Arnold to get P_R, channel image G is transformed by inverse Zigzag to get P_G, channel image B is transformed by inverse row and column to get P_B, and finally three color channel images are merged to get decrypted image.

4. Experimental Results and Analysis

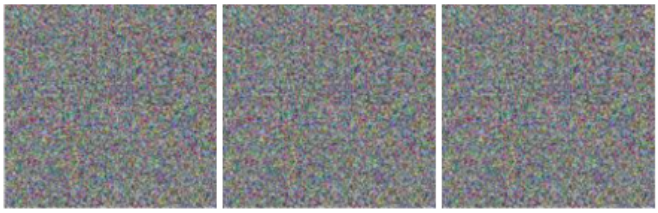
Based on Pycharm 202.1 1.1 platform, three standard test images of 512×512 TIF format color Lena, Baboon and Peppers are selected for simulation test. Firstly, the proposed algorithm is used to encrypt the test image. The three color plaintext images and corresponding ciphertext images are shown in Figure 5. The encrypted image presents random noise distribution and no meaningful information can be obtained visually, indicating that the random effect of the encryption algorithm can ensure the security of ciphertext images.

4.1. Key Space Analysis

Key space refers to the value space of the key of the encryption algorithm. In order to resist brute force attacks, the key space should be at least 2^{100} to ensure the security of encryption [11]. In the proposed encryption algorithm, 156-bit key is required as the initial value to generate two chaotic sequences, so the corresponding key space is 2^{312}. The parameters of each neuron in the neural network, such as weight, bias and other information, can be used as a key needed to encrypt an image. Since the weight of each neuron ranges from negative infinity to positive infinity, in order to facilitate the calculation of the key space, the weight of each neuron is quantified to a space of 256 sizes, so that the increased encryption complexity of each neuron is 2^8. The number of neurons used for training in this paper is 10, and each neuron has two weight information. Therefore, the key space expanded by 2^{160} after neural network randomization, corresponding to the increased key space of the two sequences is 2^{320}. Therefore, the total key space of this algorithm is 2^{632}.

The advantage of adding GAN is that the processing of neural network is equivalent to an extra layer of key protection. If the attacker cannot provide the training set, learning rate and the initial weight set by the neural network during random initialization, it is impossible to obtain the same neural network, and then the correct scrambling and diffusion

(a) Original image



(b) Encrypted image

(c) Decrypted image

Fig. 5. Three test images and corresponding ciphertext images

sequence cannot be obtained, and finally the decryption of ciphertext images cannot be realized. At the same time, the number, weight, bias and other information of neurons in the neural network will affect the size of the key space. The more neurons, the larger the key space, the better the performance of resisting violent attacks. Table 1 shows the key space comparison of several existing color image encryption algorithms. It can be seen that the key space of the proposed algorithm is higher than that of other methods. Therefore, the proposed algorithm has a better ability to resist brute force attacks.

Table 1. Key space comparison with different algorithms

Item	Reference [36]	Reference [37]	Reference [38]	GAN-DNADE
Key space calculation	$(10^{14})^4 \times 10^8$	$(10^{16})^7 \times 10^{15} \times 10^{17}$	$(2^{52})^{10}$	$(2^{156})^2 \times 2^{320}$
Key space size	$\approx 2^{535}$	$\approx 2^{515}$	2^{520}	2^{632}

4.2. Histogram Analysis

The histogram of the image is mainly used to reflect the probability of each gray level in the image, and is the main index to evaluate the ability of statistical analysis. The original image distribution has obvious features, and the image encryption system should do its best to eliminate these features and make the histogram of the encrypted image gently distributed as much as possible.

Figure 6 shows the histogram of pixel distribution of Lena image before and after encryption. Compared with the original image, the histogram distribution of the encrypted three-channel image is uniform, indicating that the frequency of different pixels in the ciphertext image is very close, and the distribution law of image pixels is difficult to analyze, thus ensuring the ability of the algorithm to resist statistical attacks and known ciphertext attacks.

4.3. Pixel Correlation Analysis

Usually digital images have strong correlations in horizontal, vertical, and diagonal directions. In order to avoid statistical attacks, encrypted images must eliminate the strong correlation of pixels in all directions to avoid attackers using channel correlation to restore the channel image and then restore the original image. In the experiment, 1000 pairs of horizontal, vertical and diagonal pixels of ciphertext three-channel images are selected to test the correlation between their adjacent pixels. When the plaintext image has a high correlation between adjacent pixels, the adjacent pixels of the plaintext three-channel image are centrally distributed on the diagonal, and the correlation coefficient is close to 1; otherwise, the value is close to 0. Table 2 compares the correlation coefficients of the plaintext images and their encrypted images of the four algorithms. Compared with other methods, the correlation coefficients of the proposed algorithm in the horizontal, vertical and diagonal directions of ciphertext images are significantly reduced, and the correlation distribution further presents a random state, indicating that the proposed algorithm has better encryption effect and scrambling performance in terms of pixel correlation.

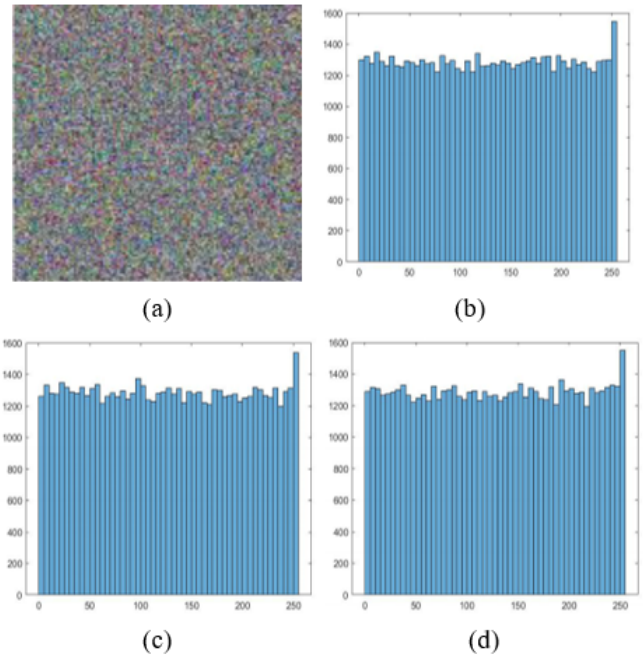


Fig. 6. Histogram analysis. (a) encrypted Lena image. (b) histogram of the red channel. (c) histogram of the green channel. (d) histogram of the blue channel

Table 2. Correlation coefficients of adjacent pixels of plaintext/ciphertext images of four algorithms

Image	Method	Horizontal	vertical	diagonal
Lena	Raw image	0.9752	0.9869	0.9627
Lena	Reference [36]	-0.0002	-0.0023	-0.0021
Lena	Reference [37]	-0.0003	0.0006	-0.0068
Lena	Reference [38]	0.0005	0.0010	0.0005
Lena	GAN-DNADE	-0.0001	-0.0014	-0.0012
Baboon	Raw image	0.9459	0.8672	0.8577
Baboon	Reference [36]	-0.0045	-0.0002	0.0001
Baboon	Reference [37]	-0.0031	-0.0002	0.0001
Baboon	Reference [38]	0.0014	0.0014	0.0029
Baboon	GAN-DNADE	-0.0013	0.0009	-0.0034
Peppers	Raw image	0.9649	0.9678	0.9572
Peppers	Reference [36]	-0.0008	-0.0067	0.0069
Peppers	Reference [37]	-0.0048	0.0052	0.0037
Peppers	Reference [38]	0.0045	0.0002	0.0025
Peppers	GAN-DNADE	-0.0032	-0.0001	-0.0031

4.4. Information Entropy Analysis

Information entropy is one of the important indicators to measure system uncertainty, which is mainly used to evaluate the unpredictability of color images. If the image pixels with gray level of 256 are evenly distributed, the maximum theoretical information entropy is 8.

In order to test the ability of the proposed algorithm to resist entropy attack, Lena, Baboon and Peppers were selected for the test in the experiment. The three images were encrypted and the information entropy was calculated respectively. The average information entropy of the three-channel ciphertext image is shown in Table 3. It can be seen that the average information entropy of the color channel images R, G and B of the ciphertext image is close to the ideal value 8, indicating that the encrypted image obtained by the algorithm in this paper is similar to other methods and has high uncertainty, which is sufficient to resist the information entropy attack.

Table 3. Average entropy results

Test image	Reference [36]	Reference [37]	Reference [38]	GAN-DNADE
Lena	7.9982	7.9996	7.9993	7.9997
Baboon	7.9979	7.9991	7.9988	7.9995
Peppers	7.9975	7.9992	7.9994	7.9995

4.5. Analysis of Anti-attack Capability

Selective plaintext attack among various attack types has obviously stronger cracking ability. If the encryption algorithm can resist selected-plaintext attacks, it can also effectively resist other types of attacks.

Relevant studies verify that if a secure encryption system can effectively generate random permutation and diffusion on all-black and all-white images, it has a high ability to resist plaintext attacks. To this end, the experiment verifies the anti-selective plaintext attack capability of the algorithm by encrypting all-black and all-white images, and the results are shown in Table 4. According to the security analysis results in Table 4, the proposed algorithm can effectively generate permutation and diffusion sequences to realize image encryption, and each evaluation index is close to the theoretical value of ciphertext image security, indicating that the proposed algorithm has a high ability to resist selected plaintext attacks.

Anti-differential Attack Analysis The number of pixel change rate (NPCR) and the unified average changing intensity (UACI) are important indicators for testing differential attacks.

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%. \tag{34}$$

Table 4. Encryption security analysis of all black/all white images

security analysis	all black	all white
Horizontal correlation	0.0022	-0.0009
Vertical correlation	-0.0017	0.0008
Diagonal correlation	-0.0034	0.0018
NPCR/%	99.6202	99.6329
UACI/%	33.4552	33.4356
information entropy	7.9993	7.9994

$$UACI = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100\%. \tag{35}$$

$$D(i,j) = \begin{cases} 0, C_1(i,j) = C_2(i,j) \\ 1, C_1(i,j) \neq C_2(i,j) \end{cases} \tag{36}$$

Where $C_1(i,j)$ and $C_2(i,j)$ are the pixel values at the encrypted image (i,j) before and after randomly changing any pixel values of the image. A larger NPCR value indicates that the encryption algorithm is more sensitive to changes in the original image. Similarly, a larger UACI value also predicts a larger average change intensity of images. The experiment compares the NPCR value and UACI value of the ciphertext image after encryption and the ciphertext image generated after pixel change by randomly changing any pixel value of the image. The results are shown in Table 5. The average NPCR and UACI values of the proposed algorithm are 99.64% and 33.49% respectively. Compared with other methods, the performance of the proposed algorithm is improved to some extent, indicating that the proposed algorithm has improved the capability of anti-differential attack compared with the existing methods.

Table 5. Anti-differential attack analysis of 3 test images%

Image	Method	NPCR	UACI
Lena	Reference [36]	99.6524	33.4558
Lena	Reference [37]	99.6098	33.4567
Lena	Reference [38]	99.6574	33.4657
Lena	GAN-DNADE	99.6351	33.5158
Baboon	Reference [36]	99.6086	33.4467
Baboon	Reference [37]	99.6061	33.4583
Baboon	Reference [38]	99.6604	33.4660
Baboon	GAN-DNADE	99.6295	33.4616
Peppers	Reference [36]	99.6321	33.4519
Peppers	Reference [37]	99.5999	33.3697
Peppers	Reference [38]	99.6285	33.4639
Peppers	GAN-DNADE	99.6397	33.4781

Anti-noise Attack Analysis In order to verify the robustness of images, the experimental data of Lena ciphertext images under different intensity noise attacks are compared with those of other algorithms. The results of anti-noise attacks are shown in Table 6.

Table 6. Anti-noise attack analysis for Lena

Salt-and-pepper noise	0.01	0.05	0.10
MSE (GAN-DNADE)	84.32	408.74	803.57
PSNR (GAN-DNADE)	28.90	25.33	21.54
MSE (Reference [36])	85.47	410.02	807.97
PSNR (Reference [36])	28.82	22.01	19.06
MSE (Reference [37])	107.31	538.93	1060.71
PSNR (Reference [37])	35.67	45.93	20.77
MSE (Reference [38])	96.14	255.71	896.32
PSNR (Reference [38])	27.94	21.58	16.94

With the increase of noise intensity, the means quare error (MSE) value of Lena image gradually increases, and the PSNR value gradually decreases. The results show that with the increase of noise attack intensity, the number of error pixels in decrypted images increases gradually, and the quality of image recovery decreases. After adding salt and pepper noise, the MSE and PSNR values of the proposed algorithm are improved compared with other literatures, indicating that the proposed algorithm has better robustness against noise attacks.

Anti-clipping Attack Analysis Ulteriorly, the experimental data of Lena ciphertext images under different reduction size attacks are compared with those of other algorithms. The experimental results are shown in Table 7 and Figure 7. As can be seen from Table 7, as the clipping size increases, more information is lost in ciphertext images. The larger the value of MSE is, the smaller the value of PSNR is, indicating that the number of error pixels in the clipped and decrypted image is increasing compared with the original image, and the recovery effect is worse. At the same time, under different clipping sizes, the MSE and PSNR values of the proposed algorithm are significantly better than those of other literatures, which indicates that the proposed algorithm has better robustness against clipping attacks.

In addition to the security of the encryption algorithm, the encryption speed is also an important index in the practical application process. Tables 8,9,10 lists the time required to encrypt the three images using this algorithm. As can be seen from tables, compared with other encryption algorithms, the proposed GAN-DNADE in this paper meets the needs of fast encryption.

5. Conclusion

In this paper, a new scheme based on GAN key generation is proposed, and a plaintext related image encryption algorithm is designed according to the generated GAN key pool.

Table 7. Anti-clipping attack analysis for Lena

Clipping attack	1/16	1/4	1/2
MSE (GAN-DNADE)	322.53	1098.64	2278.10
PSNR (GAN-DNADE)	21.98	21.89	14.93
MSE (Reference [36])	345.94	1275.10	2377.42
PSNR (Reference [36])	22.75	17.08	14.37
MSE (Reference [37])	579.99	2289.91	4578.35
PSNR (Reference [37])	20.58	20.82	11.59
MSE (Reference [38])	467.25	1788.63	3678.71
PSNR (Reference [38])	19.35	19.61	11.55

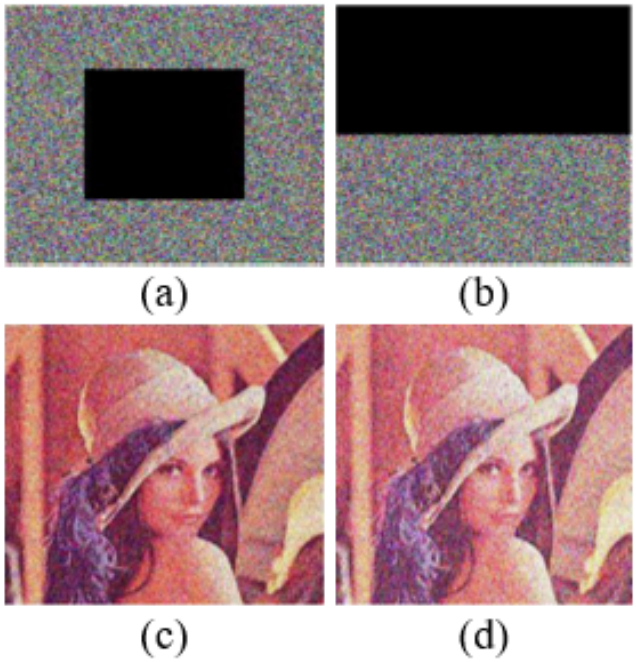


Fig. 7. The encrypted images and their decrypted images after data loss. (a) 1/16 clipping. (b) 1/2 clipping. (c) decrypted image with 1/16 clipping. (d) decrypted image with 1/2 clipping.

Table 8. Encryption time comparison for Lena

Method	size	time/s
Reference [36]	512×512	1.533
Reference [37]	512×512	1.247
Reference [38]	512×512	0.956
GAN-DNADE	512×512	0.165

Table 9. Encryption time comparison for Baboon

Method	size	time/s
Reference [36]	512×512	1.469
Reference [37]	512×512	1.188
Reference [38]	512×512	0.831
GAN-DNADE	512×512	0.154

Table 10. Encryption time comparison for Peppers

Method	size	time/s
Reference [36]	512×512	1.381
Reference [37]	512×512	1.092
Reference [38]	512×512	0.766
GAN-DNADE	512×512	0.122

Learning chaotic random keys of GAN-DNADE using GANs. The main conclusions are as follows:

1. The scheme in this paper trains and generates GAN key pool, which not only has the advantage of chaotic random key, but also has the feature of non-repeatability, which greatly improves the key generation speed and increases the security of the encryption system.
2. The random phase mask used in the algorithm based on GAN key pool is related to plaintext, so it can resist the attacks of known plaintext and selected plaintext.
3. The encryption scheme proposed in this paper can effectively improve the nonlinear characteristics of low-light level image encryption through chaotic system, so that the encryption scheme can effectively cope with various statistical analysis.
4. The key generation method is not only suitable for the image encryption algorithm proposed in this paper, but also can be applied to other encryption schemes.

References

1. Vandenbosch L, Fardouly J, Tiggemann M. Social media and body image: Recent trends and future directions[J]. Current opinion in psychology, 2022, 45: 101289.
2. S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, et al. An Anomaly Detection Model Based on Deep Auto-encoder and Capsule Graph Convolution via Sparrow Search Algorithm in 6G Internet-of-Everything[J]. IEEE Internet of Things Journal, 2024. DOI: 10.1109/JIOT.2024.3353337.
3. Jung C, Kwon G, Ye J C. Exploring patch-wise semantic relation for contrastive learning in image-to-image translation tasks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 18260-18269.
4. T. Yu, L. Chen, N. Xu, H. Xu, X. Hu and X. Zhang, "Fengyun-3E Low Light Observation and Nighttime Lights Product," in IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1-12, 2023, Art no. 4703612, doi: 10.1109/TGRS.2023.3292236.
5. Li Y, Cai F, Tu Y, et al. Low-Light Image Enhancement Under Non-uniform Dark[C]//International Conference on Multimedia Modeling. Cham: Springer Nature Switzerland, 2023: 190-201.

6. J. Liu, J. Zhang, S. Yin. Hybrid chaotic system-oriented artificial fish swarm neural network for image encryption. Evolutionary Intelligence, 16, 77-87, 2023. https://doi.org/10.1007/s12065-021-00643-5.
7. Li C, Sprott J C, Zhang X, et al. Constructing conditional symmetry in symmetric chaotic systems[J]. Chaos, Solitons & Fractals, 2022, 155: 111723.
8. Gao S, Wu R, Wang X, et al. A 3D model encryption scheme based on a cascaded chaotic system[J]. Signal Processing, 2023, 202: 108745.
9. S. Yin, H. Li, L. Teng, A. A. Laghari, V. V.a Estrela. Attribute-based Multiparty Searchable encryption model for Privacy Protection of Text Data[J]. Multimedia Tools and Applications, 2023. ttps://doi.org/10.1007/s11042-023-16818-4.
10. Gao X, Mou J, Xiong L, et al. A fast and efficient multiple images encryption based on single-channel encryption and chaotic system[J]. Nonlinear dynamics, 2022, 108(1): 613-636.
11. Zhou S, Wang X, Zhang Y, et al. A novel image encryption cryptosystem based on true random numbers and chaotic systems[J]. Multimedia Systems, 2022: 1-18.
12. M. Ji'e, D. Yan, S. Sun, F. Zhang, S. Duan and L. Wang, "A Simple Method for Constructing a Family of Hamiltonian Conservative Chaotic Systems," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 8, pp. 3328-3338, Aug. 2022, doi: 10.1109/TCSI.2022.3172313.
13. Z. Zhang, H. A. Hamadi, E. Damiani, C. Y. Yeun and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," in IEEE Access, vol. 10, pp. 93104-93139, 2022, doi: 10.1109/ACCESS.2022.3204051.
14. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
15. Gao N, Xue H, Shao W, et al. Generative adversarial networks for spatio-temporal data: A survey[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(2): 1-25.
16. Amin M, Shah B, Sharif A, et al. Android malware detection through generative adversarial networks[J]. Transactions on Emerging Telecommunications Technologies, 2022, 33(2): e3675.
17. Sekar J G, Periyathambi E, Chokkalingam A. Hybrid chaos-based image encryption algorithm using Chebyshev chaotic map with deoxyribonucleic acid sequence and its performance evaluation[J]. International Journal of Electrical & Computer Engineering (2088-8708), 2023, 13(6).
18. Wang Y, Cui W, Tao Y. A color image chunking encryption algorithm based on DNA and compound chaotic system[J]. Multimedia Tools and Applications, 2023: 1-21.
19. Li H, Zhang L, Cao H, et al. Hash Based DNA Computing Algorithm for Image Encryption[J]. Applied Sciences, 2023, 13(14): 8509.
20. Wang Z, Xu M, Zhang Y. Review of quantum image processing[J]. Archives of Computational Methods in Engineering, 2022, 29(2): 737-761.
21. Fang P, Liu H, Wu C, et al. A block image encryption algorithm based on a hyperchaotic system and generative adversarial networks[J]. Multimedia Tools and Applications, 2022, 81(15): 21811-21857.
22. X. Zhang, D. Gu, T. Wang and Y. Huang, "Old School, New Primitive: Toward Scalable PUF-Based Authenticated Encryption Scheme in IoT," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4569-4582, Dec. 2023, doi: 10.1109/TCAD.2023.3286260.
23. X. Zhang, D. Gu, T. Wang and Y. Huang, "Old School, New Primitive: Toward Scalable PUF-Based Authenticated Encryption Scheme in IoT," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4569-4582, Dec. 2023, doi: 10.1109/TCAD.2023.3286260.
24. Singh K N, Singh A K. Towards integrating image encryption with compression: A survey[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(3): 1-21.

25. Zhang D, Shafiq M, Wang L, et al. Privacy-preserving remote sensing images recognition based on limited visual cryptography[J]. CAAI Transactions on Intelligence Technology, 2023. https://doi.org/10.1049/cit2.12164.
26. Yu J, Xie W, Zhong Z, et al. Image encryption algorithm based on hyperchaotic system and a new DNA sequence operation[J]. Chaos, Solitons & Fractals, 2022, 162: 112456.
27. Ouguissi H, Saadi S, Merrad A, et al. Hybrid scheme for safe speech transmission based on multiple chaotic maps, watermarking and Arnold scrambling algorithm[J]. Multimedia Tools and Applications, 2023, 82(1): 327-346.
28. Wang X, Zhao M, Feng S, et al. An image encryption scheme using bit-plane cross-diffusion and spatiotemporal chaos system with nonlinear perturbation[J]. Soft Computing, 2023, 27(3): 1223-1240.
29. Wei C, Li G. A selective image encryption scheme using LICC hyperchaotic system[J]. IET Image Processing, 2022, 16(12): 3342-3358.
30. J. -W. Lee et al., "Privacy-Preserving Machine Learning With Fully Homomorphic Encryption for Deep Neural Network," in IEEE Access, vol. 10, pp. 30039-30054, 2022, doi: 10.1109/ACCESS.2022.3159694.
31. Sideris A, Sanida T, Dasygenis M. A Novel Hardware Architecture for Enhancing the Keccak Hash Function in FPGA Devices[J]. Information, 2023, 14(9): 475.
32. Wu, B., Zhang, T., Mao, L.: Large-scale Image Classification with Multi-perspective Deep Transfer Learning. Computer Science and Information Systems, Vol. 20, No. 2, 743-763. (2023), https://doi.org/10.2298/CSIS220714015W.
33. Zhang G, Xu Y, Rauf M, et al. Breaking the Limitation of Elevated Coulomb Interaction in Crystalline Carbon Nitride for Visible and Near-Infrared Light Photoactivity[J]. Advanced Science, 2022, 9(21): 2201677.
34. Lu L, Ren W X, Wang S D. Fractional Fourier transform: Time-frequency representation and structural instantaneous frequency identification[J]. Mechanical Systems and Signal Processing, 2022, 178: 109305.
35. Yan, Z., Hongle, D., Lin, Z., Jianhua, Z.: Personalization Exercise Recommendation Framework based on Knowledge Concept Graph. Computer Science and Information Systems, Vol. 20, No. 2, 857-878. (2023), https://doi.org/10.2298/CSIS220706024Y.
36. Yu F, Kong X, Chen H, et al. A 6D fractional-order memristive Hopfield neural network and its application in image encryption[J]. Frontiers in Physics, 2022, 10: 847385.
37. Kaur G, Agarwal R, Patidar V. Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation[J]. The Visual Computer, 2022, 38(3): 1027-1050.
38. A. Sambas et al., "A Novel 3D Chaotic System With Line Equilibrium: Multistability, Integral Sliding Mode Control, Electronic Circuit, FPGA Implementation and Its Image Encryption," in IEEE Access, vol. 10, pp. 68057-68074, 2022, doi: 10.1109/ACCESS.2022.3181424.

Xi Wang (1980-), is with the School of Artificial Intelligence and Software Engineering, Nanyang Normal University. Research direction is computer application and AI.

Received: March 14, 2024; Accepted: June 29, 2024.