IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 December 2023, accepted 22 February 2024, date of publication 29 February 2024, date of current version 15 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371888

RESEARCH ARTICLE

A Novel Medical Image Encryption Scheme Based on Deep Learning Feature Encoding and Decoding

BOFENG LONG[1], ZHONG CHEN[1,2], TONGZHE LIU[1], XIMEI WU[1], CHENCHEN HE[1], AND LUJIE WANG[1]

[1]College of Computer Science and Technology, Hengyang Normal University, Hengyang 421002, China
[2]Hunan Provincial Key Laboratory of Intelligent Information Processing and Application, Hengyang 421002, China

Corresponding author: Zhong Chen (chenzhong@hynu.edu.cn)

This work was supported in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 19A066, and in part by the Science and Technology Innovation Program of Hunan Province under Grant 2016TP1020.

ABSTRACT Medical image encryption is essential to protect the privacy and confidentiality of patients' medical records. Deep learning-based encryption, which leverages the nonlinear characteristics of neural networks, has emerged as a promising new method for protecting medical images. In this paper, we present insights into deep learning-based medical image encryption and propose a novel end-to-end medical image encryption scheme based on these insights that leverages feature encoding and decoding for encrypting and decrypting images. Firstly, we explore a method that combines keys generated by the Logistic Map with encoded plaintext image features to improve network diffusion performance. Secondly, we employ a reversible neural network to enhance plaintext image reconstruction while maintaining encryption effectiveness. Finally, we propose a series of novel loss functions to measure the cost with the ideal cryptographic algorithm and continuously optimize our network. Experimental results demonstrate that our scheme improves the performance of image encryption and decryption and resists brute force attacks, statistical attacks, noise and cropping attacks.

INDEX TERMS Medical image encryption, deep learning, convolutional neural network (CNN), feature encoding and decoding, feature fusion, logistic map.

I. INTRODUCTION

The Internet of Medical Things (IoMT) is a rapidly growing field in healthcare that employs the Internet of Things (IoT) technologies in the medical field to improve patients' and doctors' access to healthcare information and enhance the quality of care. With the development of IoMT, many medical images need to be easily distributed among doctors and patients [1]. Therefore, medical image encryption technology is essential to protect the confidentiality and integrity of medical images during storage and transmission, and it should also be able to effectively resist various network attacks, including cropping attacks and noise attacks, ensuring the quality of medical images throughout the transmission process [2]. By employing different encryption algorithms and keys, medical image encryption ensures that only authorized individuals, such as doctors and medical staff, can access the images and use them for diagnosis or treatment, safeguarding sensitive information from unauthorized access or interception. Due to the inherent characteristics of digital images, such as data redundancy, two-dimensional spatial distributions, and unequal energy distributions, traditional cryptosystems are not suitable for image encryption. In the past decade, many medical image encryption algorithms have been proposed to meet security requirements [3].

A significant proportion of image encryption research has focused on chaotic systems. These methods leverage the inherent properties of chaos, such as sensitivity to the initial conditions and parameters, mixing, and unpredictability, to provide secure image encryption [4]. They offer a

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Wang.

combination of speed, complexity, high security, reasonable computational overhead, and computational power consumption. Many chaotic systems, including high and one-dimensional systems, have been proposed for image encryption. High-dimensional chaotic systems, such as the Rössler system and the Lorenz chaotic system, are commonly employed in image encryption. Yang et al. [5] proposed a medical image encryption technique using exclusive OR (XOR) with Josephus traversing and cat mapping on the basis of the Lorenz chaotic system and SHA-512. Additionally, some researchers have also proposed new high-dimensional chaotic systems to enhance cryptographic security. Wang and Wang [6] proposed a new 6-D hyperchaotic system and employed bit-level permutation and DNA encoding to strengthen the security of the cryptosystem. However, although high-dimensional chaotic systems may offer increased security, their complexity can pose challenges for circuit implementation or encryption efficiency. On the other hand, one-dimensional chaotic systems are vulnerable to attack based on phase space reconstruction [7]. To address these limitations, some researchers have proposed enhancing one-dimensional chaotic maps by incorporating additional methods. Sun and Chen [8] proposed a method for enhancing the security of image encryption by employing a one-dimension chaotic system (logistic map) and an improved Arnold algorithm for image permutation and diffusion. However, the outputs of these chaotic systems are non-uniform. Pak et al. [9] introduced a chaotic system by using a difference of the output sequences of two same existing one-dimension chaotic maps. Improvement based on a one-dimensional chaotic system has also been continuously proposed in [10], [11], and [12]. Although some methods based on chaotic system encryption have specific security, Wang et al. [13] have pointed out that specific chaotic-based image encryption methods are also susceptible to being attacked by known plaintext methods based on deep learning.

Recently, significant advancements have been made in deep learning, particularly in image classification, image segmentation, image style transfer, and image generation. Deep learning approaches have emerged as efficient and secure image encryption methods by leveraging the non-linearity of deep neural networks and eliminating the need to manually design complex algorithms, thereby facilitating the development of more efficient and robust encryption schemes. By leveraging the learning capabilities of deep neural networks, these approaches can acquire the capacity to encode images in a manner that poses significant challenges for unauthorized decoding, ensuring the adequate protection of sensitive information [14]. In addition, neural networks have characteristics such as ultra-fast parallel processing and operate in matrix form, which is extremely well suited to image encryption.

In order to take advantage of these excellent properties of neural networks, Wang and Zhang [15] proposed a deep neural network whose weight matrices are composed of some discrete cosine transform (DCT) coefficient matrices scrambled by the logistic map and recovered using a symmetrically structured neural network and two proposed algorithms, FISTA and AD-LPMM. Man et al. [16] introduced a double image encryption algorithm that utilizes a chaotic matrix as a convolutional neural network (CNN) kernel for image fusion and encryption, focusing on enhancing security. Besides, neural networks are also used to generate random sequences. Maniyath and Thanikaiselvan [17] demonstrated the method of generating random sequences using stack autoencoder and applied them to mix encryption processing. Patel et al. [18] combined the hybrid chaos map and neural network to build a random number generator. Ding et al. [19] proposed a key generation network based on a generative adversarial network (GAN), which transferred the medical image to the private key and obtained a ciphertext image by XOR with the plaintext image. Singh et al. [20] enhanced the encryption effect by combining the GAN-based generated key with further scramble and diffusion. Apart from GAN, Zhou et al. [21] presented a color image encryption system that used a long short-term memory (LSTM) network to train chaotic signals and applied them to encrypt color images. Abdellatef et al. [22] utilized the image features extracted by CNN as the initial states of chaotic systems, and the generated sequences were used for image encryption. In addition, some methods [23], [24], [25], [26], [27] to enhance encryption security have also been proposed. However, these image encryption algorithms do not fully exploit the potential of deep learning, which only employs neural networks as an auxiliary tool within their encryption methods rather than end-to-end learning, where neural networks directly learn the encryption process from input images and encryption keys.

Therefore, certain studies have proposed deep learning-based end-to-end image encryption methods. Chai et al. [28] made full use of the advantages of CNN and GAN through compressed sensing and deep learning-based denoising. Chen et al. [29] also utilized deep learning-based CNN denoiser to improve the algorithm's robustness by improving the reconstructed images' resolution. Furthermore, some CycleGAN-based network models have been proposed, which consider ciphertext image as a stylized transfer of plaintext image. Ding et al. [30] proposed DeepEDN, which employed a Cycle-GAN as the primary learning network to transfer medical images from their original domain into a target domain for encryption and generated multiple different keys to encrypt the images due to the difficulty of training GAN. However, Bao and Xue [31] demonstrated the limited diffusion performance of CNN when directly inputting the original image, failing to generate encrypted images that meet diffusion metrics. To improve safety, they proposed an improved method that concatenated the plaintext image with random images as inputs into the encryption network, performed further diffusion and outputted the result as a ciphertext image to improve the avalanche effect. Subsequently, some improvements have been proposed

in [32] and [33]. In addition to GAN-based methods, Sang et al. [7] proposed a novel image encryption method based on logistic chaotic systems and deep autoencoder, which scrambled the plaintext image using a logistic chaotic system and then encoded the scrambled image with a deep autoencoder to generated the ciphertext image. Zhu et al. [34] proposed an end-to-image image diffusion model, FEDRes-Net, and a novel loss function based on pixel entropy to enhance the security of ciphertext images. To strengthen the security of the encrypted network, Li and Peng [35] introduced an attention mechanism to enhance the model's response to the region of interest within the medical image. However, these algorithms do not exhibit stronger enough resistance against cropping and noise attacks for transmitting in complex networks.

Previous research explored the application of deep learning to image encryption and made significant advancements in the field. However, these studies ignored the importance of the loss functions and did not provide relevant insights for the network model. In this study, we have critically analyzed the limitations of existing algorithms and models and proposed novel loss functions and models to address these shortcomings. The main contributions of this work are summarized as follows:

(1) We proposed some guidelines for image encryption using CNN and built a novel self-supervised model based on these guidelines. In contrast to other models, our model required only the plaintext image for training and can reduce the security problem caused by privacy leakage. Experimental results showed that our model could improve the quality of image encryption and effectively solve the problem of the cropping attack and noise attack.

(2) We proposed a set of novel loss functions to guide the encryption process toward achieving the desired encryption metric. This proposal addressed the issue that current deep learning-based end-to-end image encryption methods struggle to effectively encrypt images and are difficult to optimize. By leveraging the unique characteristics of image encryption, our proposed loss functions enabled the network to learn the main features of the plaintext image while ensuring the security of the encrypted image. To demonstrate their effectiveness, we conducted ablation experiments which confirmed the efficacy of our proposed loss functions.

(3) Different from previous encryption methods, our method did not need to transmit network parameters as encryption and decryption keys, and users only need to input encrypted images and keys to restore images, reducing the number of keys that need to be saved during the encryption and decryption process.

The remainder of this article is organized as follows. In Section II, we introduce the chaotic system and the architecture utilized in our network. Section III presents our proposed loss function and describes the training process. In Section IV, we evaluate the performance of our scheme through experimental results. Finally, in Section V, we provide a brief conclusion and discuss potential avenues for future research.

II. PREPARATORY WORK

A. CHAOTIC SYSTEM

1) LOGISTIC MAP

Logical map, due to it efficient execution time relative to other chaotic maps, is widely employed in fields such as image encryption. We employed it to generate *feature keys* to encrypt the features of plaintext images in this paper, and the formula is shown in Eq. (1),

$$x_n = \mu x_{n-1}(1 - x_{n-1}), \tag{1}$$

where x_n denotes to state variable at the n-th moment, μ lies within the closed interval from 3.5699 to 4 ($3.5699 \leq \mu \leq 4$), and x belongs to between 0 and 1 ($0 < x < 1$). Depending on the choice of μ, the Logistic Map can exhibit chaotic dynamical behaviors. Although the corresponding sequences $\{x_0, x_1, \ldots, x_n\}$ can be considered chaotic, there exist certain isolated ranges of μ that show non-chaotic behavior. As illustrated by the bifurcation diagrams in Figure 1, when μ exceeds 3.5699, the sequences are limited in the range of 0 to 1, with stabilization occurring around 3.8284.

2) GENERATE FEATURE KEYS

The *feature key* is the key generated by the Logistic Map in the key generator that is used to encrypt the features of the plaintext image. In order to generate sufficiently sensitive feature keys, we introduced SHA-256 to ensure that different images will produce completely inconsistent control parameters, even if they differ by only one pixel. The specific process is shown in Figure 2, The initial parameters $x_0^1, x_0^2, x_0^3, x_0^4$ is obtained through Eq. (2) and generate the feature keys of the same size as the plaintext features, respectively.

$$\begin{cases} x_0^1 = \dfrac{\bmod\,(h(p:p+64)+b_1, 2^{64})+1}{2^{64}+2}, \\ x_0^2 = \dfrac{\bmod\,(h(p+65:p+128)+b_2, 2^{64})+1}{2^{64}+2}, \\ x_0^3 = \dfrac{\bmod\,(h(p+129:p+192)+b_3, 2^{64})+1}{2^{64}+2}, \\ x_0^4 = \dfrac{\bmod\,(h(p+193:p+256)+b_4, 2^{64})+1}{2^{64}+2}, \\ p = \bmod\,(\sum_{i=1}^{WH}(\mathrm{im}(i)\times i), 256), \end{cases} \tag{2}$$

where b_i is the user-defined key, the plaintext image im undergoes a SHA-256 hashing process, yielding a 256-bit hash value h. This value is subsequently partitioned into four parts starting from position p, each containing 64 bits.

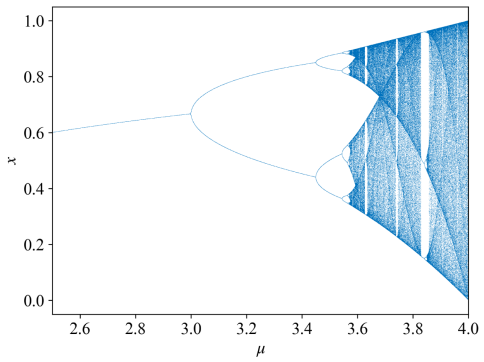


FIGURE 1. The bifurcation diagram for logistic map with initial value x_0 of 10^{-5}, the points associated with each value of μ represent a set of chaotic sequences.

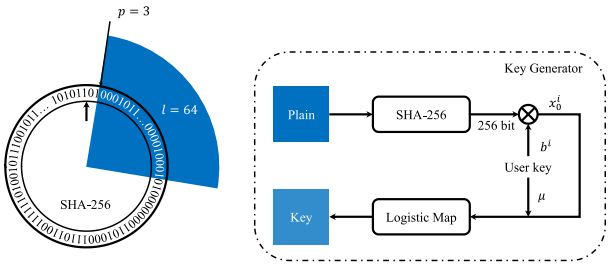


FIGURE 2. The process of generating keys by the key generator. The plaintext image generates a 256 bit key by SHA-256, which is processed and summed with b^i to get x^i, and used as parameters of the logistic map together with μ.

B. NETWORK ARCHITECTURE

Convolutional neural networks are highly effective in extracting and compressing image features, making them popular in image compression and generation. In this paper, we proposed a novel convolutional neural network that forms an encrypted image by encoding plaintext image features, fusing them with feature keys, and restoring the image by decrypting the encrypted features. To enhance the encryption performance of the network, we proposed specific guidelines:

(1) Since the plaintext image changes by one pixel, the pixels of the corresponding encrypted image must be completely changed, which requires ensuring that the receptive field corresponding to the ciphertext image must be larger than or equal to the input plaintext image.

(2) To resist the noise or data loss attack during network transmission, we can simulate these attacks by adding a *dropout* function, which randomly changes certain pixel values to some fixed constant to ensure that the loss of part of the ciphertext image will not affect the overall decryption performance.

(3) The generated ciphertext image must comply with the image encryption standard, and the recovered image must be within a specific acceptable range.

The network is mainly composed of the Feature Encoder Module and Feature Decoder Module (FEM/FDM), Feature Fusion Encryption Module (FFEM), Feature Fusion Decryption Module (FFDM), and Feature Fusion Passing Module (FFPM), the structure of which is illustrated in Figure 3.

1) FEATURE ENCODER MODULE & FEATURE DECODER MODULE

The Feature Encoder Module (FEM) mainly consists of a three-layer convolution (a convolution layer and a Res Block layer) and a LeakyReLU activation function (it solves the problem of zero gradient in the negative region of ReLU, which facilitates backpropagation), which encodes the main features of the plaintext image through convolution. After passing the Feature Encoder Module, the resulting feature map has half the width and height, but twice the number of channels, compared to the previous feature map. However, each channel will extract the contour and texture information of the plaintext image separately and represent them compressed. Correspondingly, the Feature Decoder Module (FDM) consists of three layers of transposed convolution and a LeakyReLU activation function, which will reduce the decrypted feature maps, and the corresponding feature maps will be doubled in width and height and halved in channels. In the last layer of the encryption process, we used the sub-pixel convolution (Pixel Shuffle) to recover the size of the original image. As with StyleGAN2 [36], we removed the regularization to reduce the blob-shape artifacts that resemble water droplets. The network in this paper has four FEM and FDM, where the feature map of the plaintext image is repeatedly encoded and decoded four times. This process enables the encrypted and decrypted images to satisfy the encryption and decryption metrics.

2) FEATURE FUSION PASSING MODULE

The Feature Fusion Passing Module (FFPM) comprises multiple Res blocks that dynamically fuse plaintext image features and feature keys to form a new feature map. This fusion ensures the preservation of features while improving the overall diffusion performance of the ciphertext image, as shown in Figure 4 (b). The Res block associated with the plaintext features expands the receptive field of the ciphertext image, eliminating the need for downsampling to 1×1 to achieve the same receptive field as the original image size. In contrast, the Res block associated with the feature key projects a random sequence onto an appropriate distribution and maintains the same dimension between the feature key and the feature map, ensuring encryption in the same dimension. The Res block in the middle of Figure 4 (b) is used to dynamically fuse the plaintext image feature F_i with the feature key K_i, which generates new dynamic fused parameters α_i ($0 < \alpha < 1$) by inputting the stacked plaintext image features and key features. The dynamic fusion is defined as follows Eq. (3),

$$F_{i+1} = \alpha_i F_i + (1 - \alpha_i) K_i \,. \tag{3}$$

The dynamic allocation of K_i and F_i through the network ensures that the features are encrypted in this layer and can still be reconstructed in the next layer.

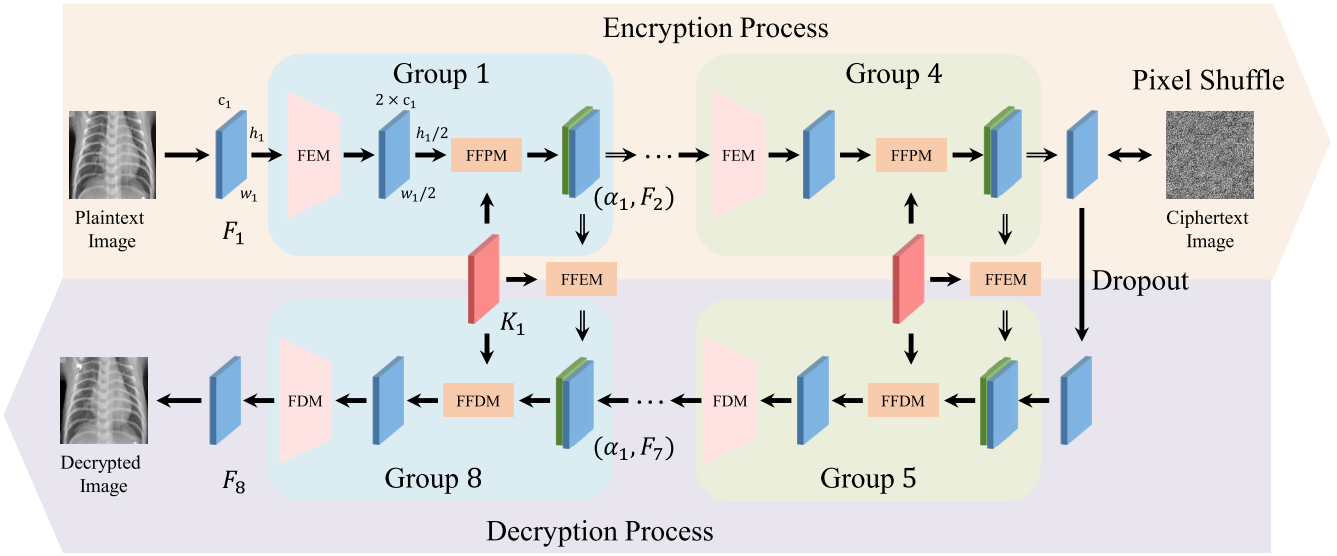


FIGURE 3. The architecture of our model. The plaintext image and the key are dynamically fused to form an encrypted image, and the plaintext image is recovered by dynamic decoupling.

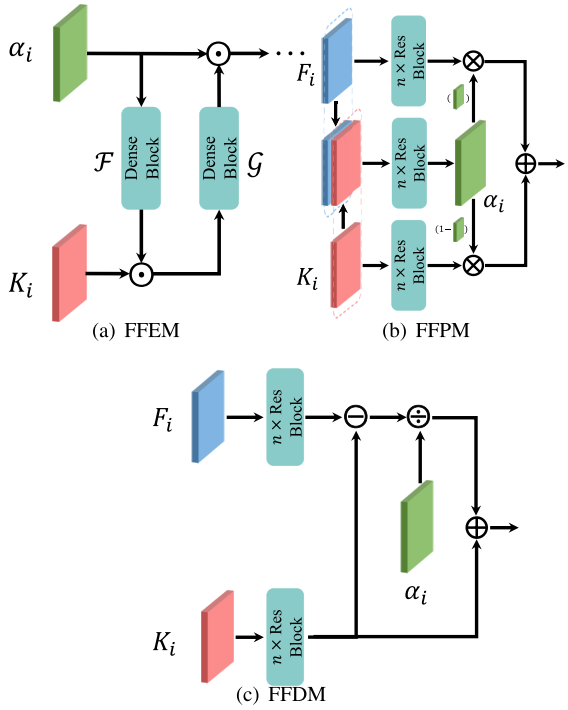


FIGURE 4. (a) Feature Fusion Encoding Module, (b) Feature Fusion Passing Module and (c) Feature Fusion Decryption Module. The FFEM is composed of a series of Dense Blocks, and the FFPM or FFDM is comprised of various Res Blocks.

3) FEATURE FUSION ENCRYPTION MODULE

The Feature Fusion Encryption Module (FFEM) is inspired by the Reversible Residual Networks (RevNets), which consists of a series of reversible blocks and is a variant of an invertible neural network, as shown in Figure 4 (a). During encryption, $\odot$ represents an addition operation, the formula is shown in Eq. (4), while it represents a subtraction operation during decryption.

$$\begin{cases} EK_i = K_i + \mathcal{F}(\alpha_i) \\ EP_i = \alpha_i + \mathcal{G}(EK_i), \end{cases} \tag{4}$$

where $\mathcal{F}(\bullet)$ and $\mathcal{G}(\bullet)$ represents the Dense Block on the Figure 4 (a), these residual functions $\mathcal{F}$ and $\mathcal{G}$ analogous to those in standard ResNets. The function for image reconstruction is as follows Eq. (5),

$$\begin{cases} \alpha_i = EP_i - \mathcal{G}(EK_i) \\ K_i = EK_i - \mathcal{F}(F_i). \end{cases} \tag{5}$$

This network can represent volume-preserving mappings since it is invertible and has unit determinant Jacobian, which is widely used in image steganography. The detail of RevNet can be found in [37]. By leveraging this network, the correlations within the data can be effectively obfuscated, thereby enhancing the overall security of the dynamic fused parameters α_i. The module guarantees the accurate transfer of corresponding data to the decryption module since it is designed to be reversible.

4) FEATURE FUSION DECRYPTION MODULE

The Feature Fusion Decryption Module (FFDM) attempts to separate the original image features from the fused features, as shown in Figure 4 (c). Since FFEM is a RevNet, it can output decrypted dynamic fused parameters by inputting the corresponding feature key and the encrypted dynamic fused parameter in the decryption phase. These decrypted parameters are fed into FFDM together with the features recovered by the network to restore the features. The recovery

TABLE 1. The parameters and structure of the network model.

	Number	Kernel Size	Channel		Parameters	Total Parameters
			Input	Output		
conv	1	5×5	1	32	832	832
deconv	1	3×3	32	32	9,248	9537
	1	3×3	32	1	289	
conv + Res Block (FEM)	1	3×3	32	64	92,352	3,706,944
	1	3×3	64	128	369,024	
	1	3×3	128	256	1,475,328	
	1	3×3	256	256	1,770,240	
deconv + Res Block (FDM)	1	3×3	256	256	1,770,240	2,545,056
	1	3×3	256	128	590,208	
	1	3×3	128	64	147,648	
	1	3×3	64	32	36,960	
Res Block (FFPM)	4×2	3×3	64	64	73,856	20,920,896
	1	3×3	128	64	118,976	
	3×2	3×3	128	128	295,168	
	1	3×3	256	128	475,520	
	3×2	3×3	256	256	1,180,160	
	1	3×3	512	256	1,901,312	
	3×2	3×3	256	256	1,180,160	
	1	3×3	512	256	1,901,312	
Res Block (FFDM)	3×2	3×3	256	256	1,180,160	19,253,120
	1	3×3	256	256	1,180,160	
	3×2	3×3	256	256	1,180,160	
	1	3×3	256	256	1,180,160	
	3×2	3×3	128	128	295,168	
	1	3×3	128	128	295,168	
	4×2	3×3	64	64	73,856	
	1	3×3	64	64	73,856	
Dense Block (FFEM)	2	3×3	64	64	3,319,680	32,966,784
	2	3×3	128	128	5,753,088	
	2	3×3	256	256	11,947,008	
	2	3×3	256	256	11,947,008	

process is defined as follows Eq. (6),

$$F_{i+1} = \frac{F_i - K_i}{\alpha_i} + K_i\,, \tag{6}$$

where F_i represents the feature obtained from the preceding layer of the network, K_i denotes the key entered in the corresponding encryption phase and α_i is the dynamic fused parameter acquired in the FFPM. When all values of α_i are 1, the network functions as an autoencoder network.

The corresponding network parameters are shown below in Table 1. In the encryption stage, we performed four downsamplings, each downsampling contains one FEM and one FFPM, and the number of residual blocks in the FFPM is 4, 3, 3, and 3, which can ensure that the obtained receptive field is greater than or equal to the original features. In the decryption stage, the number of residual blocks in the corresponding FFPM was 3, 3, 3, and 4, because the corresponding Group are mirror symmetric.

III. EXPERIMENTS

A. OVERVIEW

This paper introduces a novel end-to-end medical image encryption and decryption network based on deep learning. The network utilized the convolutional neural network's characteristics to extract image features. During the encryption phase, the plaintext image features were encoded and encrypted to ensure secure transmission. In the decryption phase, the correct keys were inputted into the network to restore the original image features and recover the plaintext image. Figure 5 illustrates the encryption and decryption process, where the user provides the plaintext image and key parameters to the encryption network and key generator, respectively. The key generator generates a series of feature keys for the encryption network, which are fused with the plaintext image features to generate feature maps and encrypted images that meet the image encryption metrics. Upon receiving the encrypted feature maps, the authorized

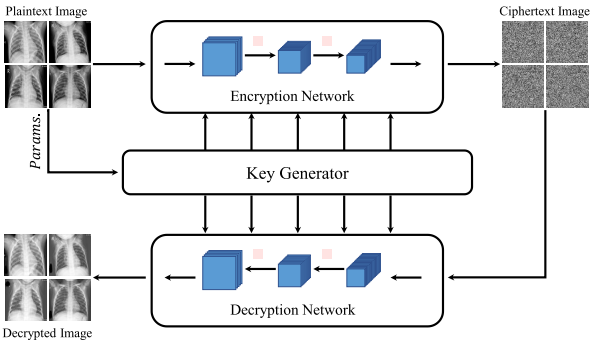


FIGURE 5. Flowchart for image encryption and decryption. The plaintext image and the key are dynamically fused to form an encrypted image, and the plaintext image is recovered by dynamic decode.

user can utilize the decryption network to decouple these feature maps from the keys and generate decrypted images by entering the corresponding key parameters from the encryption phase. If an incorrect key is used, the decryption network will produce a different image compared to the one obtained during the encryption phase.

B. LOSS FUNCTION

A proper loss function can guide the network to converge quickly. In deep learning-based image encryption, the non-differentiability of image entropy poses a challenge. To address this issue, Zhu et al. [34] introduced a loss function based on the entropy of the individual pixels in the encryption image. This approach reduced the amount of information contained in the image pixels. Additionally, Sang et al. [7] proposed a uniform distribution loss, which ensured a uniformly distributed histogram for encryption images. We dedicate this section to exploring a range of loss functions specifically designed for deep learning-based image encryption, aiming to overcome the limitations of previous metrics for evaluating encrypted images.

1) HISTOGRAM AND ENTROPY LOSS

Although the image histogram is non-differentiable, it is possible to approximate a differentiable histogram. Ustinova and Lempitsky [38] proposed a histogram loss to match positive and negative sample pairs. This approach assumes that a point c, situated halfway between points a and b, has a probability of $(b-c)/(b-a)$ at point a and $(c-a)/(b-a)$ at point b. For an image that uses a floating-point representation, it is not only possible to calculate the histogram of the image by accumulating its probability at the corresponding floating-point pixel, but also to instruct the network to move point c closer to a or b, which makes the obtained image more accurate. To ensure a uniformly distributed histogram for the encrypted image, the histogram loss is defined as in Eq. (7),

$$\mathcal{L}_{hist} = \frac{1}{N}\sum_{x=0}^{N-1}\left(H(x) - \frac{W \times H}{N}\right)^2, \tag{7}$$

where $H(x)$ is the differential histogram of the image obtained by the described method in the vicinity of point x, while N is the number of equal-width bins in the histogram. Additionally, W and H are the width and height of the image, respectively. Suppose the histogram of the image is uniformly distributed, its value is a constant of $(W \times H)/N$. The loss quantifies the discrepancy between the image and the ideal histogram, with a value closer to zero indicating a more uniformly distributed histogram.

In addition to the histogram, the information entropy should reflect the consistency of the grayscale distribution, with higher values indicating greater consistency. In an ideal case of a random gray image, the information entropy would be equal to eight, the formula is defined as in Eq. (8),

$$\text{ENTROPY} = -\sum_{x=0}^{N-1}\overline{H(x)}\log_2\overline{H(x)}, \tag{8}$$

where $\overline{H(x)}$ is the normalization of $H(x)$, which is also the probability density function of the image. The corresponding entropy loss would be close to zero, which can be defined as shown in Eq. (9),

$$\mathcal{L}_{ent} = 8.0 - \text{ENTROPY}, \tag{9}$$

the value of $\mathcal{L}_{ent}$ close to zero indicates that the image contains less information and has higher uncertainty.

Although both the histogram loss and entropy loss can make the histogram uniformly distributed, the histogram loss can move the floating-point value close to the effective position, and information entropy loss can effectively assist the histogram loss to decline in backpropagation. The corresponding losses is described as $\mathcal{L}_{stats} = \lambda_{hist}\mathcal{L}_{hist} + \lambda_{ent}\mathcal{L}_{ent}$, where λ_{hist} is set to 1 and λ_{ent} is set to 0.125.

2) CORRELATION LOSS

Adjacent pixel correlation refers to the degree of similarity between neighboring pixels in an image, which measures how much the value of a pixel is dependent on the values of its neighboring pixels. The correlation coefficients in horizontal, vertical, and diagonal directions are calculated as Eq. (10),

$$r_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{10}$$

here,

$$E(x) = \frac{1}{N}\sum_{i=0}^{N-1}x_i, \tag{11}$$

$$D(x) = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - E(x))^2, \tag{12}$$

$$\text{cov}(x,y) = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - E(x))(y_i - E(y)). \tag{13}$$

where y is the adjacent pixel of x, r_{xy} is the correlation between two adjacent pixels and N is the number of chosen pixel pairs. We selected all pixels of the image for training and 10000 pixels randomly for evaluation. The loss function of $\mathcal{L}_{cor}$ is defined as in Eq. (14), it added the absolute value operation for r_{xy} only to prevent the correlation from tending to be negative in training.

$$\mathcal{L}_{cor} = \|r_{xy}\|_1. \tag{14}$$

When the image becomes more monochromatic or the figure of the adjacent pixel tends to be X-shaped, the correlation coefficient also tends towards zero, which is undesirable for encrypted images. To address this issue, we proposed a differentiable 2D-Histogram Loss that can effectively solve the homogeneity problem. By using this

loss, the encrypted image can retain its desired characteristics and meet the encryption requirements.

The x-axis and y-axis of the 2D histogram represent the pixel values of the selected image and its neighboring pixels, respectively. Each cell in the histogram represents the count of pixel pairs (x, y) consisting of the corresponding x pixel and its neighboring y pixels, providing a visual representation of the correlation between the data points. The construction of the 2D histogram loss function can be defined mathematically, as shown in Eq. (15),

$$\mathcal{L}_{2d} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \left(H(x, y) - \frac{W \times H}{N^2} \right)^2 , \tag{15}$$

where N represents the number of intervals of the histogram, which constructs an approximate histogram of neighboring pixels $H(x, y)$ and uses it as an auxiliary for correlation loss. To balance memory usage and backpropagation efficiency, we often set the value of N to 8. In this paper, we used $\mathcal{L}_c = \lambda_{cor}\mathcal{L}_{cor} + \lambda_{2d}\mathcal{L}_{2d}$ to represent the correlation loss, where λ_{cor} and λ_{2d} are both set to 1.

3) DIFFUSION LOSS

An effective encryption method should exhibit strong diffusion performance, which means that changing one pixel in the plaintext image results in a completely unpredictable change in the ciphertext image. The diffusion performance is commonly evaluated using metrics such as the number of pixels change rate (NPCR) and unified average changing intensity (UACI). The formula of NPCR is expressed in Eq. (16),

$$\text{NPCR} = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} D(i, j) , \tag{16}$$

here,

$$D(i, j) = \begin{cases} 0 , & \text{if} \quad C_1(i, j) = C_2(i, j) \\ 1 , & \text{if} \quad C_1(i, j) \neq C_2(i, j), \end{cases} \tag{17}$$

where $D(i, j)$ describes the difference between encrypted images C_1 of the original image and encrypted images C_2 with some pixels changed from the original image. In general, a high NPCR value close to 1 indicates that the encrypted images corresponding to different plaintext images are distinct. The loss function is represented in Eq. (18),

$$\mathcal{L}_{NPCR} = 1.0 - \text{NPCR} , \tag{18}$$

while the NPCR metric evaluates the number of pixel values that change during the differential attacks. However, the two images cannot just be simple pixel inconsistency, if the pixel values of encrypted image C_1 can be changed into those of encrypted image C_2 through a modular addition operation, then the two encrypted images are not essentially different. Therefore, the UACI metric is usually used to measure the average rate of pixel changes in the corresponding positions of the two images. Accordingly, the formula of UACI is expressed in Eq. (19),

$$\text{UACI} = \frac{1}{W \times H \times N} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \| C_1(i, j) - C_2(i, j) \| . \tag{19}$$

The paper [39] shows that the ideal UACI value should be close to 33.4635% for a 256 × 256 grayscale image. Consequently, the corresponding loss function is as follows in Eq. (20),

$$\mathcal{L}_{UACI} = \| \epsilon - \text{UACI} \|_1 , \tag{20}$$

where ϵ can be set to the desired UACI ideal value. The diffusion performance of the network can be improved by combining NPCR loss and UACI loss and applying the following network architecture. In this paper, we used $\mathcal{L}_{diff} = \lambda_N \mathcal{L}_{NPCR} + \lambda_U \mathcal{L}_{UACI}$ to represent the diffusion loss, with both λ_N and λ_U set to 0.1.

4) SIMILARITY LOSS

In addition to existing methods, we also used the cosine similarity loss as the encryption and decryption loss of the reversible neural network. During the encryption phase, we must ensure that the fused parameters differ from the original parameters, as indicated by the loss function in Eq. (21),

$$\mathcal{L}_{disim} = \sum_{i=1}^{N} \max\Big(\epsilon, \cos\big(\alpha_i, \text{INNE}(\alpha_i)\big) \Big) , \tag{21}$$

where cos denotes the cosine similarity function, ϵ denotes the similarity threshold, which is usually set to zero, N denotes the number of corresponding groups, α_i refers to the dynamic fused parameters of the plaintext image at layer i, and INNE represents the forward process of the reversible neural network. The loss is mainly to make the input parameters not similar to the output parameters. Furthermore, we employed the L2 paradigm of fused parameters and random noise to make them mutually different from the original parameters. In contrast, the loss function of Eq. (22) will be used in the decryption process to recover the original parameters,

$$\mathcal{L}_{sim} = \sum_{i=0}^{N-1} \Big(1 - \cos\big(\alpha_i, \text{INNR}(EP_i)\big) \Big) , \tag{22}$$

where EP_i denotes the i-th layer of encrypted parameters using the reversible neural network, and INNR denotes the reverse process of the reversible neural network, which reverts the parameters to the original parameters.

We imposed constraints on the decryption network to prevent the accurate recovery of the image using incorrect keys. By constructing an $m \times n$ loss matrix $\mathcal{M}$, where each row represents a different encrypted image and each column represents a different encryption key, the element on the main diagonal of the matrix corresponds to the correct key associated with the current ciphertext image, and its

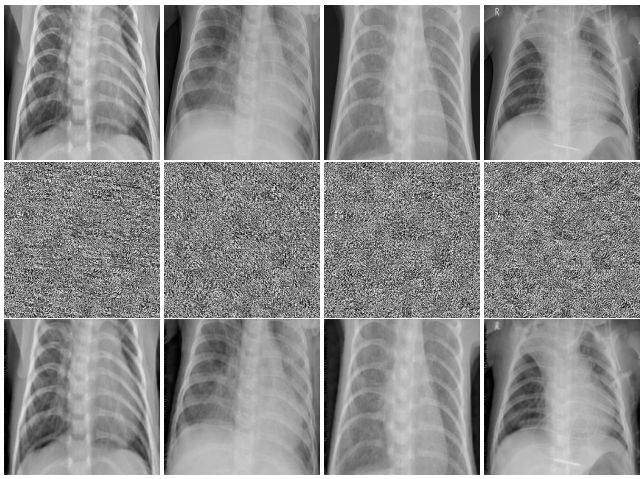


FIGURE 6. The encryption and decryption results of the images. From top to bottom as the plaintext image, ciphertext image, and decrypted images.

corresponding value should be 0. The remaining keys, unable to restore the plaintext image correctly, are assigned values equal to the sum of the losses of the decrypted image $\hat{x}$ and various factors, such as the L1 paradigm loss of random noise, statistical loss, and diffusion loss. Consequently, the loss corresponding to an incorrect key can be obtained in Eq. (23),

$$\mathcal{L}_{error} = \sum_{i=1,j=1}^{m,n} \left((\mathbf{1} - I) \cdot \mathcal{M}_{ij} \right), \tag{23}$$

where I denotes the unit diagonal matrix, $\mathbf{1}$ represents a matrix filled with ones, and $\mathcal{M}_{ij}$ is the loss value of the ciphertext image in the i-th row and j-th column key of the matrix $\mathcal{M}$.

C. TRAINING

1) PRE-TRAINING

Pre-training of the network is essential to enhance feature encoding and decoding capabilities and facilitate accurate restoration of the plaintext image while expediting the training process. Initially, we removed the fusion operation with feature keys in FFPM to ensure that the features of plaintext images can be comprehensive compression, allowing the entire network to effectively encode and decode these features while exhibiting appropriate generalization capabilities. The pre-training process resembles that of SRGAN [40], both employing perceptual loss and content loss as the training losses for the generator. Concurrently, co-training of the generator and discriminator takes place through adversarial loss. However, in contrast to the application of image super-resolution, our approach solely focuses on efficient image reconstruction.

2) ENCRYPTION AND DECRYPTION

The pre-trained model serves as the foundation for image encryption. Within the i-th group as depicted in Figure 3, the plaintext image feature F_i and the feature key K_i are fused to form the fused feature F_{i+1}. However, these fused features do not conform to a uniform distribution. Consequently, to protect the plaintext feature from potential attacks, each fused feature F_i undergoes repeated fusions in the next layer, which ensures that the attacker cannot discern these features during network transmission.

During the image recovery process, the encrypted image must be restored. Initially, the fused parameters α_i output from the encryption stage is input into the decryption network together with the feature key K_i. Subsequently, the ciphertext image feature F_i recovered from the previous network layer is de-fused with the feature key and restored to the current ciphertext feature F_{i+1}. As the process of reduction and de-fusion progresses, the original image is ultimately restored with high accuracy.

Consequently, the loss function employed during the encryption process is denoted as Eq. (24),

$$\mathcal{L}_{encrypted} = \lambda_s \mathcal{L}_{stats} + \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_{diff} + \lambda_{cos} \mathcal{L}_{disim}. \tag{24}$$

Similarly, the loss function employed during the decryption process is represented as Eq. (25),

$$\mathcal{L}_{decrypted} = \lambda_e \mathcal{L}_{error} + \lambda_{sim} \mathcal{L}_{sim} + \lambda_{l1} \left\| \hat{x} - x \right\|_1, \tag{25}$$

where $\hat{x}$ denotes the recovered image and x represents the plaintext image. Throughout the training process, the weight assigned to λ_s, λ_c, λ_d and λ_{cos} are 1, 1, 0.1, and 0.25, respectively. Additionally, the weights assigned to λ_e, λ_{sim} and λ_{l1} are 1, 0.5, and 1, respectively.

We view the encryption network as a fusion between the feature distribution F of the image and the random uniform distribution U to generate a new data distribution. As this distribution propagates forward, it becomes constrained by the loss function, yielding a new uniform distribution. Conversely, the decryption network disentangles the feature distribution of encrypted images from the known random distribution U, thereby restoring the original data distribution.

The proposed scheme is implemented with the Pytorch framework on a computer equipped with Nvidia RTX 3090 GPU in the Windows 10 operating system. We used Python 3.10 to code our proposed model and conducted training and testing using both the Tuberculosis Chest X-rays (Montgomery) dataset [41] and OCT & Chest X-Ray images dataset [42]. All images in our experimental data sets are preprocessed to a size of 256 × 256. The initial global learning rate was set to 0.0001 and decreased with

TABLE 2. Evaluation result of encrypted and decrypted images in SSIM and PSNR.

	Ref. [30]	Ref. [34]	Ref. [31]	Ours
SSIM (encryption)	0.014	0.009	0.007	0.009
PSNR (encryption)	N/A	8.076	7.477	7.798
SSIM (decryption)	0.913	0.950	0.936	0.945
PSNR (decryption)	36.34	31.96	33.18	30.99

TABLE 3. Ablation experiment for the proposed loss function.

Loss	Entropy	Correlation			NPCR	UACI
		Horizontal	Vertical	Diagonal		
$\mathcal{L}_{hist}$	7.9965	0.02901	-0.00928	-0.13413	98.40%	17.09%
$\mathcal{L}_{ent}$	7.9965	-0.13093	0.01694	-0.02665	98.41%	17.55%
$\mathcal{L}_{cor}$	7.9690	-0.00489	0.00281	0.00154	98.68%	17.67%
$\mathcal{L}_{2d}$	7.9883	0.01494	-0.00522	-0.00038	98.25%	16.70%
$\mathcal{L}_{stats}$	7.9971	-0.27750	-0.34624	0.19780	98.25%	16.70%
$\mathcal{L}_{c}$	7.9723	0.00095	0.00052	0.00023	99.11%	25.33%
$\mathcal{L}_{diff}$	7.9750	-0.00381	0.01016	-0.01563	99.44%	33.47%
$\mathcal{L}_{stats} + \mathcal{L}_{c}$	7.9967	0.00320	-0.00011	-0.00019	98.71%	22.41%
$\mathcal{L}_{stats} + \mathcal{L}_{diff}$	7.9917	0.02245	0.01795	-0.09793	99.42%	31.55%
$\mathcal{L}_{diff} + \mathcal{L}_{c}$	7.9740	0.00064	0.00116	0.00048	99.52%	33.38%
$\mathcal{L}_{stats} + \mathcal{L}_{c} + \mathcal{L}_{diff}$	7.9852	-0.00072	-0.00185	-0.00033	99.50%	32.75%

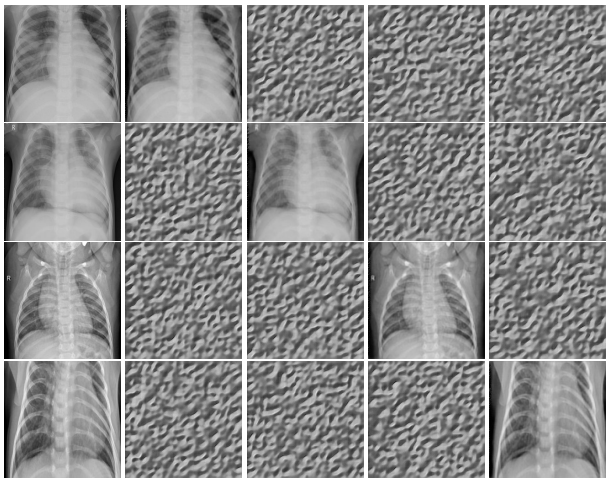


FIGURE 7. The plaintext image and its corresponding decrypted image. (a) The plaintext image; (b)-(e) representing the result of decryption using different key: K_1, K_2, K_3, and K_4.

a decay rate of 0.9 for each epoch. The batch size was set to four, and the AdamW optimization algorithm optimized the network parameters. We trained it for four days. Due to the randomness of the key and the instability of GAN, it is impossible to train an identical model. Meanwhile, it is also possible to generate more different network models by replacing the appropriate chaotic sequences.

D. EXPERIMENT ANALYSIS

1) EXPERIMENT RESULT

The results of encrypting and decrypting several chest X-ray images using our model are presented in Figure 6. Our encryption and decryption method demonstrated its efficacy by effectively encrypting the images, producing encrypted images significantly different from the original images, and accurately restoring them during decryption. Table 2 compares the structural similarity (SSIM) and peak signal-to-noise ratio (PSNR) between the encrypted image, decrypted image, and original images, demonstrating the strong performance of our encryption and decryption method.

SSIM is a measure of the similarity between two images that take into account their brightness, contrast, and structure. The formula for SSIM is given by Eq. (26),

$$SSIM(x, y) = \left[\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right]^{\alpha} \cdot \left[\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right]^{\beta} \cdot \left[\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}\right]^{\gamma}, \quad (26)$$

where μ_x and μ_y, σ_x and σ_y are the mean and standard deviation of x and y, respectively, σ_{xy} is the covariance of x and y; and C_1, C_2, C_3 and α, β, γ are constants. The SSIM value is closer to 1 when the two images are more similar, and PSNR is expressed as the ratio of the maximum power sum of an image to the maximum accuracy of the noise it can represent, and its formula is shown in Eq. (27),

$$\text{PSNR} = 10\log_{10}\left(\frac{I_{\max}^2}{\text{MSE}}\right), \quad (27)$$

where $I_{\max}$ denotes the maximum value of the image, which is 255 if the image range is between [0, 255], and MSE denotes the mean square error between the two images. Usually, when the PSNR is greater than 30 dB, it is difficult for the human eye to detect the difference between the restored and original images.

Moreover, the network will produce the wrong decrypted image when the wrong image key is input, and the result is shown in Figure 7. The image demonstrates that only the correct key enables the complete restoration of the image.

2) ABLATION EXPERIMENT

To verify the effectiveness of our method, we conducted ablation experiments on both the proposed loss functions and the model architecture. In the loss ablation experiments, we maintained the network structure while modifying specific components of the loss functions and trained these models accordingly. The results of the ablation experiment for the loss function are shown in Table 3. As the similarity loss is solely used in the FFEM to reconstruct the original image, we did not conduct ablation experiments for this loss.

TABLE 4. Comparative with other methods in information entropy, correlation, and diffusion metric.

	Entropy	Correlation			NPCR	UACI
		Horizontal	Vertical	Diagonal		
Original Image	7.2737	0.9766	0.9900	0.9792	N/A	N/A
Ref. [7]	7.9695	-0.0402	0.0584	-0.0048	N/A	N/A
DeepEDN [30]	7.9500	0.5112	0.3881	0.4575	94.21%	N/A
Ref. [31]	7.7675	-0.0050	0.0011	-0.0022	1.5%	0.006%
Ref. [31][d]	7.9972	0.0004	0.0005	0.0011	99.64%	33.49%
FEDResNet [34]	3.2024	N/A	N/A	N/A	24.26%	12.70%
FEDResNetSD [34][d]	7.9963	-0.0018	N/A	N/A	99.64%	33.49%
Ref. [33]	7.9937	-0.0041	-0.0278	-0.0067	99.63%	33.55%
AT-ResNet-CM [35]	7.9965	0.0010	N/A	N/A	N/A	N/A
Ours	7.9949	0.0014	-0.0007	0.0002	99.55%	33.02%

[d] represents this method take further diffusion or scrambling operator.

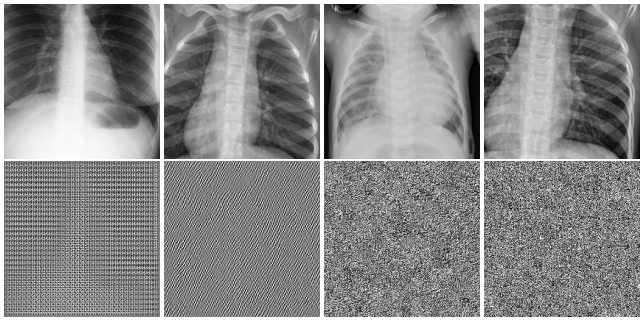


FIGURE 8. Encryption results using different models. (a) The plaintext image and ciphertext image without the key fusion operation of the FFEM; (b) The plaintext image and ciphertext image with only three groups of modules; (c) The plaintext image and ciphertext image with only four groups of modules; and (d) The plaintext image and ciphertext image with only five groups of modules.

Upon the table, it can observe that $\mathcal{L}_{stats}$ and $\mathcal{L}_c$ can improve the information entropy of encrypted images and diminish the correlation between neighboring pixels, respectively. However, the $\mathcal{L}_{diff}$ may inhibit the increase of information entropy due to its larger gradient during backpropagation. We need to reduce the weight of $\mathcal{L}_{diff}$ at the beginning of training and increase it appropriately once the $\mathcal{L}_{stats}$ reaches a lower value.

In our study, we also performed ablation experiments on individual modules to assess the effectiveness of our model while adhering to the established design guidelines, shown in Figure 8. Figure 8 (b) represents the number of groups in Figure 3 network structure was reduced from four to three during the encryption and decryption, while the number of convolutions per layer was increased. This resulted in the retention of some plaintext image contours in the encrypted image, which may be attributed to the insufficient compression of the plaintext image by the three-group configuration of the network. Conversely, when the number of groups exceeded four, the encrypted image exhibited improved effectiveness, albeit at the expense of a substantial increase in model size. Furthermore, our results indicated that omitting the fusion operation with feature keys in the FFEM did not significantly enhance the diffusion performance of the encrypted image.

We also conducted a comparative analysis with several deep learning-based, end-to-end image encryption methods. As demonstrated in Table 4, our approach significantly improved the performance metrics without additional diffusion or scrambling techniques. Furthermore, the incorporation of our proposed loss, as described in Ref. [34], further enhances these metrics. However, the method described in Ref. [34] is limited by its downsampling and restoration of the original image size during the encryption process, which results in an inability to restore regions affected by cropping attacks. On the other hand, our method can solve this problem.

IV. EVALUATION

A. KEY SPACE ANALYSIS

A secure algorithm must possess a key space of adequate size to increase its resistance against brute force attacks. The key space refers to the total number of distinct keys involved in the encryption and decryption processes. In our case, the key was generated by the logistic chaotic system, which required the initial parameter $x \in (0, 1)$. When expressed in double-precision floating-point type, this allows for the representation of a minimum of 2^{61} (the number of bits in the corresponding ranges of x accordingly to IEEE 754 rules) different parameters. As we required four different keys, our parameters contained approximately $(2^{61})^4 = 2^{244}$ distinct key combinations. Hence, our encryption algorithm exhibited an exceptionally large key space, significantly mitigating the vulnerability to extensive brute-force attacks.

B. STATISTICAL ANALYSIS

Statistical analysis is a crucial tool for evaluating the security of image encryption algorithms. By conducting statistical analysis, it can be demonstrated that an encryption algorithm is capable of withstanding general statistical attacks and safeguarding the security of the image. One important evaluation metric in image statistical analysis is the image histogram, which reflects the distribution of pixels within an image. For instance, as depicted in Figure 9, the histogram (b)

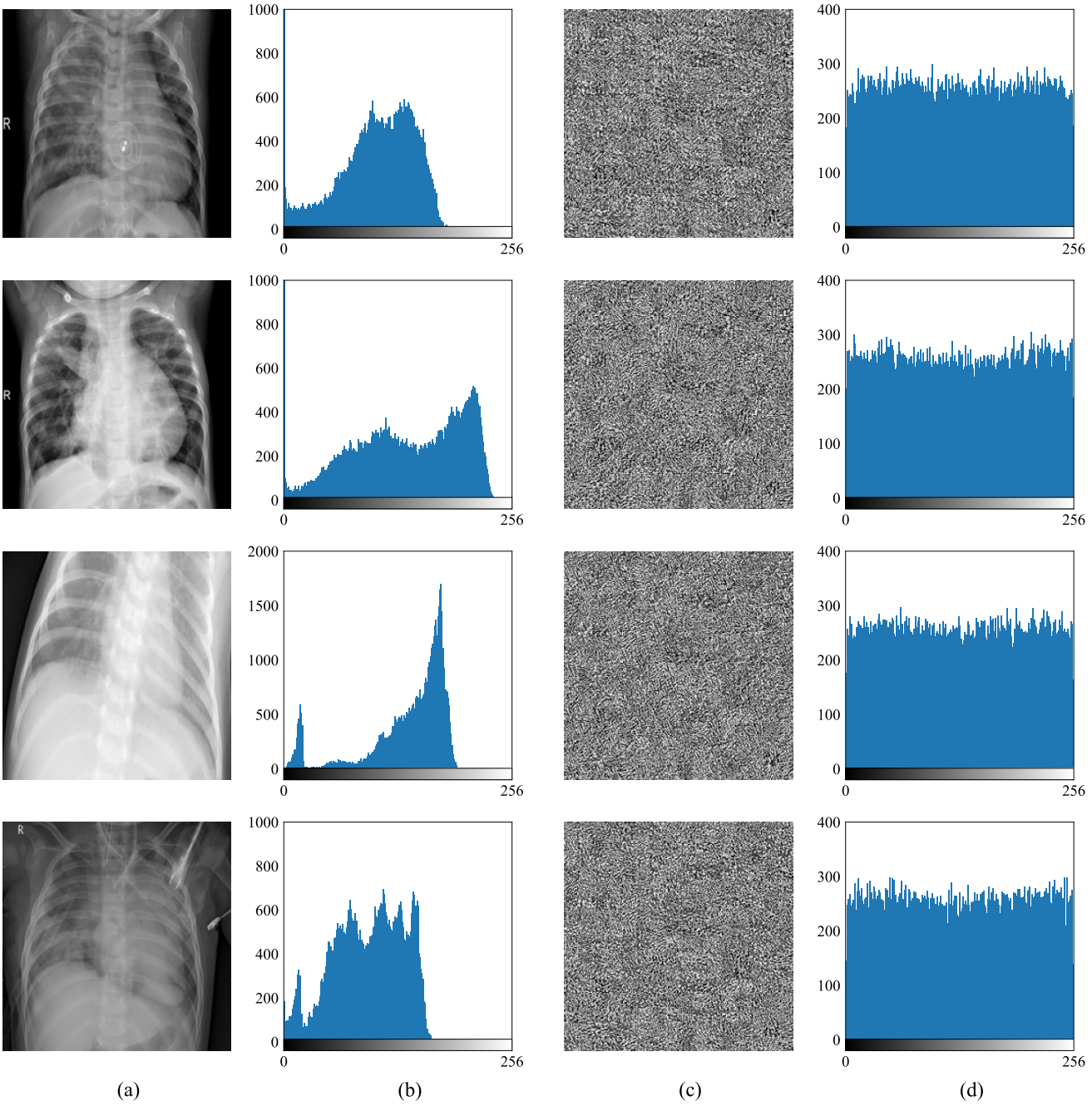


FIGURE 9. Histogram of plaintext and ciphertext images. (a) Plaintext Image; (b) Histogram of Plaintext Image; (c) Ciphertext Image; (d) Histogram of Ciphertext Image.

corresponding to the plaintext image (a) reveals that the image contains a higher proportion of white and black pixels. An encryption algorithm should ensure that the histogram of the ciphertext image differs from that of the plaintext image and that the histograms of different ciphertext images are consistent to prevent attackers from inferring the specific content of an image by analyzing its pixel distribution. This is illustrated in Figure 9, where Figure 9 (c) represents the encrypted image corresponding to Figure 9 (a) and Figure 9 (d) is the histogram corresponding to Figure 9 (c). It is evident that the histograms of these encrypted images differ significantly from those of their respective plaintext images, and that their histograms are consistent.

Another method for determining the amount of information an image carries is through analysis of its information entropy, as shown in Eq. (8). A value closer to 8.0 indicates that the image carries less information. As shown in Table 4 and Figure 12, the information entropy of ciphertext images in our test set is close to 8.0, indicating that these images convey less information and further

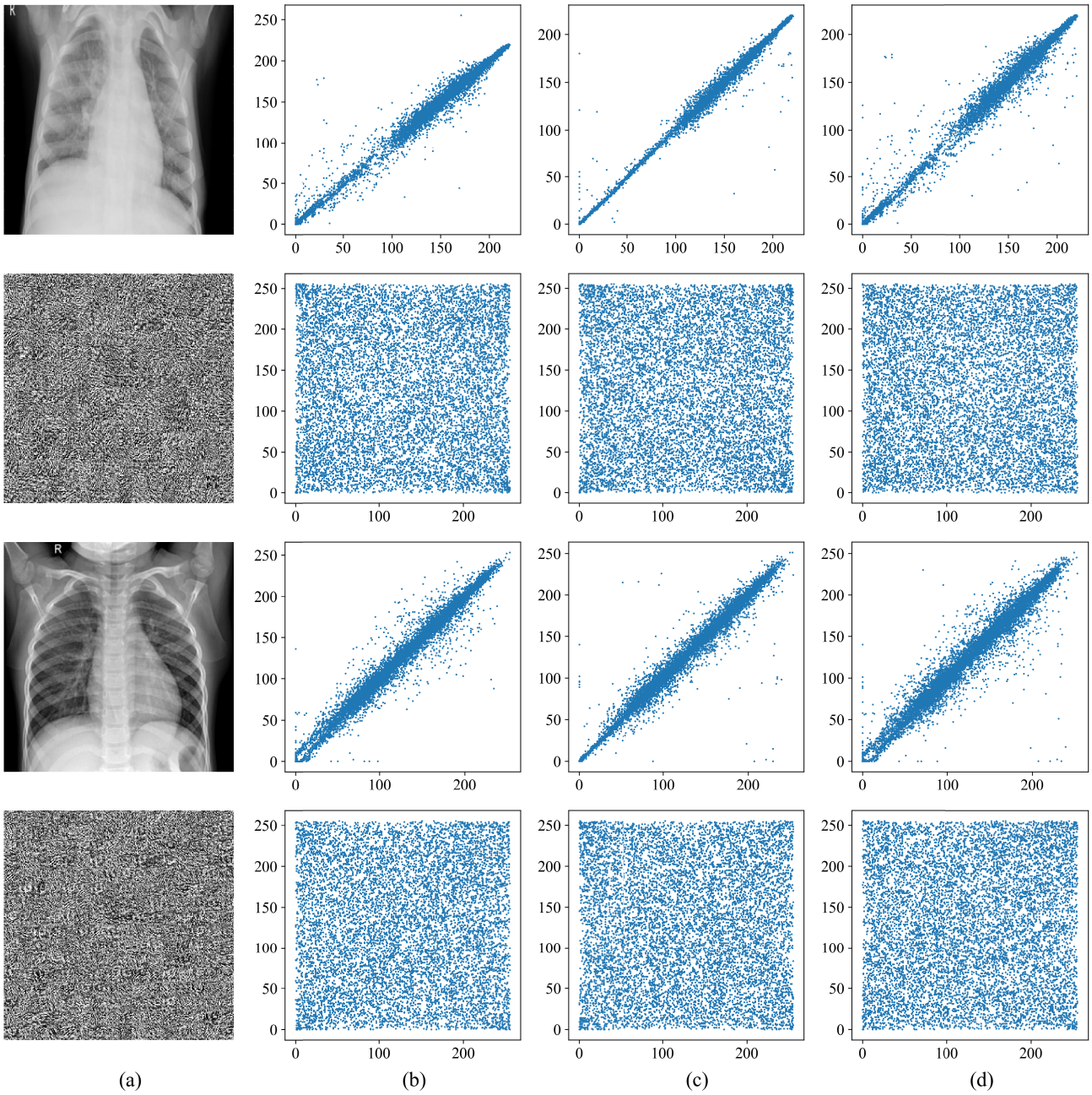


FIGURE 10. Neighboring pixel correlation between plain and ciphertext images in horizontal, vertical, and diagonal directions. (b) Horizontal adjacent pixel correlation; (c) Vertical adjacent pixel correlation; (d) Diagonal adjacent pixel correlation.

demonstrates the effectiveness and security of our encryption algorithm.

Adjacent pixel correlation is another important metric used in image statistical analysis, which reflects the degree of similarity between two adjacent pixels. A correlation value close to 1 or −1 indicates a high correlation between adjacent pixels, while a value close to zero indicates a low correlation and makes it difficult for an attacker to infer the content of an image through statistical analysis. In plaintext images, as shown in the first and third rows of Figure 10, there is often a high correlation between neighboring pixels in horizontal, vertical, and diagonal directions. If the correlation of a ciphertext image is high, an attacker may be able to approximate and restore the original image using these correlations. However, as seen in the second and four rows of Figure 10, the correlation between the corresponding ciphertext images is low, making it more difficult for an attacker to conduct a successful statistical analysis attack.

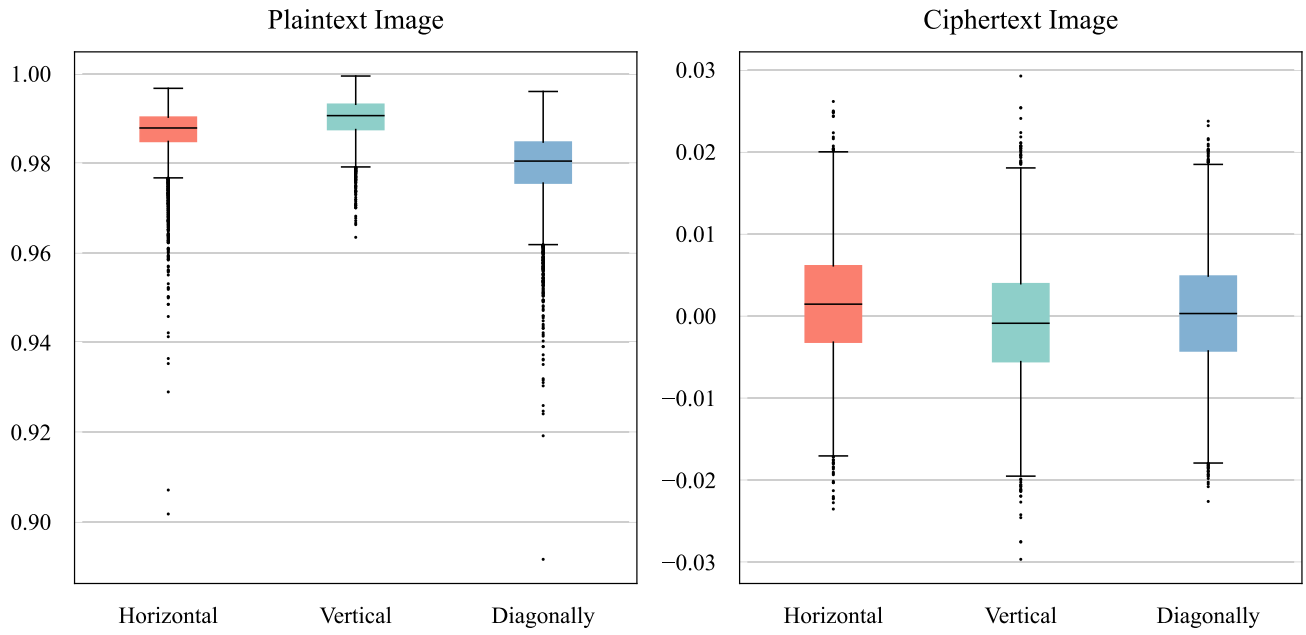


FIGURE 11. Boxplots of the correlation of neighboring pixels in horizontal, vertical, and diagonal directions for plaintext and ciphertext images in test set.

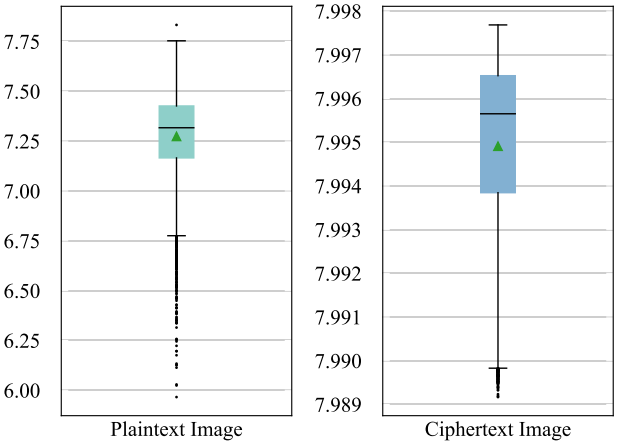


FIGURE 12. The information entropy of both plaintext and ciphertext images was evaluated on the test set, where the black line in the middle of the box is the data's midpoint, and the green triangles are the average value.

In addition, we also counted the correlations of the images in the test set, with results shown in Figure 11 and Table 4. The correlation in plaintext images is close to 1.0, while that in the ciphertext images is close to zero, indicating that our encryption algorithm can effectively resist statistical analysis attacks and provides strong encryption.

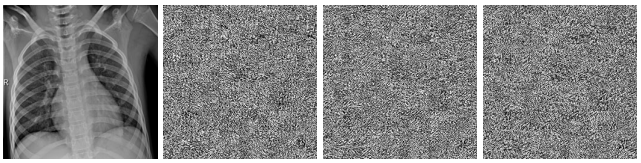
C. DIFFERENTIAL ATTACK

The diffusion performance of an encryption algorithm is a crucial criterion for evaluating its ability to resist differential attacks. When a pixel in the plaintext image is changed, all pixels in the corresponding ciphertext image should also be changed. If only some pixels in the ciphertext image change, an attacker can infer the pixels in the plaintext image by changing the pixels in the plaintext image several times and observing changes in the corresponding ciphertext image. However, if all pixels are changed, the cost of an attack increases significantly, thus securing the ciphertext image. As shown in Figure 13, we changed one pixel and 10% pixels in the plaintext image (a), resulting in encrypted images (c) and (d), respectively. Using Eq. (16) and Eq. (19), we calculated the NPCR values of 99.58% and 99.60% for the ciphertext image (b) with (c) and (d), respectively, and the UACI values of 33.69% and 33.39%, respectively. These values are close to the ideal for image encryption, indicating that our algorithm can resist differential attacks. Additionally, we also evaluated the performance of our method compared to other methods, which have good diffusion performance, as shown in Table 5 and Table 4.

TABLE 5. Comparison of NPCR and UACI corresponding to different methods after changing plaintext image pixels.

	Method	NPCR	UACI
One pixel changed	Ref. [31]	99.64%	33.49%
	Ref. [34][d]	100.0%	49.99%
	Ours	99.55%	33.02%
1% pixel changed	Ref. [30]	94.21%	N/A
	Ref. [34][d]	99.62%	49.80%
	Ours	99.58%	32.97%

In the process of image encryption, when only one value of the key is changed slightly, the corresponding ciphertext image should also have good encryption properties. In this experiment, we changed four different values of K_1 (adding 10^{-15} to each x_0), as shown in Figure 14, to obtain NPCR criterion of 99.5422%, 99.6124%, 99.5483%, and 99.6277%,

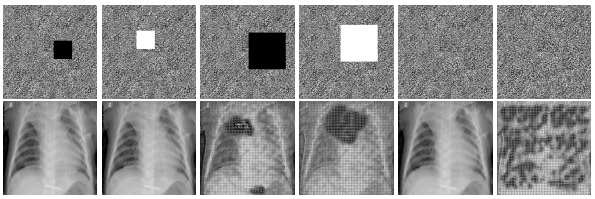


(a) (b) (c) (d)

FIGURE 13. Ciphertext image after changing pixels. (a) Plaintext Image; (b) Ciphertext Image; (c) Ciphertext Image after changing one pixel of the plaintext image; (d) Ciphertext image after changing 1% pixel of the plaintext image.

(a) (b) (c) (d) (e)

FIGURE 14. Ciphertext image with different keys. (a) Ciphertext Image with $K_1 = \{x_0^1, x_0^2, x_0^3, x_0^4\}$; (b) Ciphertext Image of $x_0^1 + 10^{-15}$ in K_1; (c) Ciphertext Image of $x_0^2 + 10^{-15}$ in K_1; (d) Ciphertext Image of $x_0^3 + 10^{-15}$ in K_1; (e) Ciphertext Image of $x_0^4 + 10^{-15}$ in K_1.



(a) (b) (c) (d) (e) (f)

FIGURE 15. Noise and cropping attack. (a) 50 × 50 black cut attack, (b) 50 × 50 white cut attack, (c) 100 × 100 black cut attack, (d) 100 × 100 white cut attack, (e) 1% gaussian noise attack, (f) 10% gaussian noise attack.

respectively, and the corresponding UACI criterion are 32.5752%, 33.2755%, 32.5435%, and 33.2661%, which proves that the algorithm is sensitive to key.

D. NOISE AND CROPPING ATTACK

When transmitting data on the network, noise and other attacks may be introduced. If the decryption process cannot remove the effects of noise, an attacker may be able to manipulate the decrypted image by adding noise or removing parts of the data. Previous deep learning-based encryption schemes have been shown to be vulnerable to cropping attacks, where the network restores location information related to the original image during upsampling, resulting in an inability to restore the content of clipped regions. However, by using a compressed feature map as the encrypted image, our scheme ensures that the restored image is unaffected by cropping attacks. Our tests have shown that our scheme can resist at least 10% of noise attacks and more than 50 × 50 of cropping attacks. As demonstrated in Figure 15 and Table 6, Our method can still restore the original image closely when the noise and cropping attacks are not severe.

TABLE 6. Evaluate the differences in PSNR and SSIM Between the plaintext image and its corresponding decrypted image under noise and data loss attack.

Attacks	PSNR	SSIM
Decryption Image	31.65	0.9402
30×30	31.64	0.9401
50×50 (black)	30.22	0.9244
50×50 (white)	30.29	0.9250
100×100 (black)	13.98	0.2056
100×100 (white)	11.08	0.1004
1% gaussian noise	31.39	0.9377
10% gaussian noise	10.01	0.0635

E. SECURITY ANALYSIS

Ciphertext-only attack: In this scenario, the attacker only has access to the ciphertext and tries to analyze the patterns in the ciphertext image through statistical analysis. However, our method generates encrypted images with histograms that are nearly uniformly distributed and exhibit low correlation, thereby preventing the attacker from determining the image content through statistical analysis.

Known plaintext attack: In this scenario, the attacker has some pairs of plaintext and ciphertext and tries to find the mapping between the plaintext and the ciphertext. However, our method uses a nonlinear network mapping to encrypt each pixel of the image, which depends on both the pixel value and the key value of the original image. Moreover, Our model has high diffusion performance with 99.55% pixel change rate between different encrypted images. Even if the attacker trains a network with plaintext-ciphertext pairs, it cannot restore the original image without knowing the complete feature map and the key.

Chosen plaintext attack: In this scenario, the attacker has some plaintexts, their corresponding ciphertexts, and the encryption model, and tries to train a decryption network to recover the original image. However, our feature fusion encryption module requires consistent parameters to decrypt the corresponding feature map, and we obtain a complete network by pre-training. If we retrain the whole network from scratch, it may not converge to the same decryption network due to the instability of GAN. Therefore, this method is also secure against this attack.

Chosen ciphertext attack: In this scenario, the attacker has some ciphertexts, their partial plaintexts, and the decryption model, and tries to determine the plaintext or key of other ciphertext images. However, this method also fails because of the same reason as above, which ensures the security of our image encryption method.

V. CONCLUSION

An end-to-end medical image encryption system based on the convolutional neural network is proposed in this paper, which uses the feature encoding and decoding capabilities of deep learning to encrypt and decrypt images. To enhance the network's diffusion performance and security, the feature keys generated by chaotic system are fused with encoded

plaintext image features. Additionally, reversible neural networks are used to encrypt dynamic fused parameters, which are then transmitted as encrypted parameters to improve reconstruction effectiveness. Statistical loss, diffusion loss, and similarity loss are proposed to guide the network in generating secure encrypted images, prevent decryption with incorrect keys, and efficiently restore encrypted images. Experiments on the Chest X-Ray dataset demonstrate our scheme's resistance to numerous attacks and reduced parameter requirements during encryption and decryption. Although our current scheme is able to achieve good encryption results, its effectiveness on color images remains untested due to some constraints. Furthermore, it is worth further studying the potential of using deep learning based schemes to generate meaningful images that are different from plaintext images, or improving encryption performance by enhancing network feature extraction capabilities.

REFERENCES

[1] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent advances in the Internet-of-Medical-Things (IoMT) systems security," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8707–8718, Jun. 2021.

[2] T. N. Lakshmi, S. Jyothi, and M. R. Kumar, *Image Encryption Algorithms Using Machine Learning and Deep Learning Techniques—A Survey*. Cham, Switzerland: Springer, 2021, pp. 507–515.

[3] H. M. Ghadirli, A. Nodehi, and R. Enayatifar, "An overview of encryption algorithms in color images," *Signal Process.*, vol. 164, pp. 163–185, Nov. 2019.

[4] M. Kaur and V. Kumar, "A comprehensive review on image encryption techniques," *Arch. Comput. Methods Eng.*, vol. 27, no. 1, pp. 15–43, Jan. 2020.

[5] N. Yang, S. Zhang, M. Bai, and S. Li, "Medical image encryption based on Josephus traversing and hyperchaotic Lorenz system," *J. Shanghai Jiaotong Univ.*, vol. 29, no. 1, pp. 91–108, Dec. 2022.

[6] T. Wang and M.-H. Wang, "Hyperchaotic image encryption algorithm based on bit-level permutation and DNA encoding," *Opt. Laser Technol.*, vol. 132, Dec. 2020, Art. no. 106355.

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

[8] X. Sun and Z. Chen, "A new image encryption strategy based on Arnold transformation and logistic map," in *Proc. 11th Int. Conf. Comput. Eng. Netw.* Singapore: Springer, 2022, pp. 712–720.

[9] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," *Signal Process.*, vol. 138, pp. 129–137, Sep. 2017.

[10] J. Tang, F. Zhang, and H. Ni, "A novel fast image encryption scheme based on a new one-dimensional compound sine chaotic system," *Vis. Comput.*, vol. 39, no. 10, pp. 4955–4983, Oct. 2023.

[11] Z. Hua, Y. Zhou, and H. Huang, "Cosine-transform-based chaotic system for image encryption," *Inf. Sci.*, vol. 480, pp. 403–419, Apr. 2019.

[12] S. Zhu, X. Deng, W. Zhang, and C. Zhu, "Secure image encryption scheme based on a new robust chaotic map and strong S-box," *Math. Comput. Simul.*, vol. 207, pp. 322–346, May 2023.

[13] F. Wang, J. Sang, C. Huang, B. Cai, H. Xiang, and N. Sang, "Applying deep learning to known-plaintext attack on chaotic image encryption schemes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 3029–3033.

[14] K. Panwar, S. Kukreja, A. Singh, and K. K. Singh, "Towards deep learning for efficient image encryption," *Proc. Comput. Sci.*, vol. 218, pp. 644–650, Jan. 2023.

[15] C. Wang and Y. Zhang, "A novel image encryption algorithm with deep neural network," *Signal Process.*, vol. 196, Jul. 2022, Art. no. 108536.

[16] Z. Man, J. Li, X. Di, Y. Sheng, and Z. Liu, "Double image encryption algorithm based on neural network and chaos," *Chaos, Solitons Fractals*, vol. 152, Nov. 2021, Art. no. 111318.

[17] S. R. Maniyath and V. Thanikaiselvan, "An efficient image encryption using deep neural network and chaotic map," *Microprocessors Microsyst.*, vol. 77, Sep. 2020, Art. no. 103134.

[18] S. Patel, V. Thanikaiselvan, D. Pelusi, B. Nagaraj, R. Arunkumar, and R. Amirtharajan, "Colour image encryption based on customized neural network and DNA encoding," *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14533–14550, Nov. 2021.

[19] Y. Ding, F. Tan, Z. Qin, M. Cao, K. R. Choo, and Z. Qin, "DeepKeyGen: A deep learning-based stream cipher generator for medical image encryption and decryption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4915–4929, Sep. 2022.

[20] O. D. Singh, S. Dhall, A. Malik, and S. Gupta, "A robust and secure immensely random GAN based image encryption mechanism," *Multimedia Tools Appl.*, vol. 82, no. 13, pp. 19693–19743, May 2023.

[21] S. Zhou, Z. Zhao, and X. Wang, "Novel chaotic colour image cryptosystem with deep learning," *Chaos, Solitons Fractals*, vol. 161, Aug. 2022, Art. no. 112380.

[22] E. Abdellatef, E. A. Naeem, and F. E. A. El-Samie, "DeepEnc: Deep learning-based CT image encryption approach," *Multimedia Tools Appl.*, vol. 83, no. 4, pp. 11147–11167, Jan. 2024.

[23] Y. He, Y.-Q. Zhang, X. He, and X.-Y. Wang, "A new image encryption algorithm based on the OF-LSTMS and chaotic sequences," *Sci. Rep.*, vol. 11, no. 1, p. 6398, Mar. 2021.

[24] Y. Liu, G. Cen, B. Xu, and X. Wang, "Color image encryption based on deep learning and block embedding," *Secur. Commun. Netw.*, vol. 2022, pp. 1–14, Oct. 2022.

[25] J. Liang, Z. Song, Z. Sun, M. Lv, and H. Ma, "Coupling quantum random walks with long- and short-term memory for high pixel image encryption schemes," *Entropy*, vol. 25, no. 2, p. 353, Feb. 2023.

[26] A. Elsonbaty, A. A. Elsadany, and W. Adel, "On reservoir computing approach for digital image encryption and forecasting of hyperchaotic finance model," *Fractal Fractional*, vol. 7, no. 4, p. 282, Mar. 2023.

[27] H. Lin, C. Wang, L. Cui, Y. Sun, C. Xu, and F. Yu, "Brain-like initial-boosted hyperchaos and application in biomedical image encryption," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8839–8850, Dec. 2022.

[28] X. Chai, Y. Tian, Z. Gan, Y. Lu, X.-J. Wu, and G. Long, "A robust compressed sensing image encryption algorithm based on GAN and CNN," *J. Modern Opt.*, vol. 69, no. 2, pp. 103–120, Jan. 2022.

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

[30] Y. Ding, G. Wu, D. Chen, N. Zhang, L. Gong, M. Cao, and Z. Qin, "DeepEDN: A deep-learning-based image encryption and decryption network for Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1504–1518, Feb. 2021.

[31] Z. Bao and R. Xue, "Research on the avalanche effect of image encryption based on the cycle-GAN," *Appl. Opt.*, vol. 60, no. 18, pp. 5320–5334, 2021.

[32] K. Panwar, A. Singh, S. Kukreja, K. K. Singh, N. Shakhovska, and A. Boichuk, "Encipher GAN: An end-to-end color image encryption system using a deep generative model," *Systems*, vol. 11, no. 1, p. 36, Jan. 2023.

[33] J. Wu, W. Xia, G. Zhu, H. Liu, L. Ma, and J. Xiong, "Image encryption based on adversarial neural cryptography and SHA controlled chaos," *J. Modern Opt.*, vol. 68, no. 8, pp. 409–418, May 2021.

[34] L. Zhu, W. Qu, X. Wen, and C. Zhu, "FEDResNet: A flexible image encryption and decryption scheme based on end-to-end image diffusion with dilated ResNet," *Appl. Opt.*, vol. 61, no. 31, pp. 9124–9134, 2022.

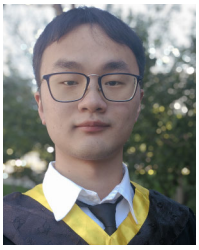
[35] X. Li and H. Peng, "Chaotic medical image encryption method using attention mechanism fusion ResNet model," *Frontiers Neurosci.*, vol. 17, 2023, Art. no. 1226154.

[36] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8110–8119.

[37] A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse, "The reversible residual network: Backpropagation without storing activations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 2214–2224.

[38] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4170–4178.

[39] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun.*, vol. 1, no. 2, pp. 31–38, 2011.
[40] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
[41] S. Jaeger, S. Candemir, S. Antani, Y. X. Wang, P. X. Lu, and G. Thoma, "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases," *Quant. Imag. Med. Surg.*, vol. 4, no. 6, p. 475, Dec. 2014.
[42] D. S. Kermany et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, Feb. 2018.



BOFENG LONG received the B.S. degree from the Computer Engineering Department, Taiyuan Institute of Technology, China. He is currently pursuing the master's degree in electronic information with Hengyang Normal University, Hunan, China. He majors in digital image encryption and deep learning.

ZHONG CHEN received the M.S. degree in applied mathematics from the Department of Applied Mathematics, Southwest Jiaotong University, China, in 2004, and the Ph.D. degree in mechanical engineering from Hunan University, China, in 2018. He is currently an Associate Professor with the School of Computer Science and Technology, Hengyang Normal University, Hunan, China. His main research interests include digital image encryption, nonlinear dynamics, and deep learning.

TONGZHE LIU received the B.S. degree from the Department of Science and Engineering, Shuda College, Hunan Normal University, China. He is currently pursuing the master's degree in electronic information with Hengyang Normal University, Hunan, China. His research interest includes deep learning for image encryption and applications.

XIMEI WU received the B.S. degree in software engineering from Hengyang Normal University, China, where she is currently pursuing the master's degree in electronic information. Her major is digital image encryption.

CHENCHEN HE received the B.S. degree from the School of Computer Science and Technology, Hengyang Normal University, Hunan, China, where she is currently pursuing the master's degree in electronic information. She majors in digital image encryption and image processing.

LUJIE WANG received the B.S. degree in computer science and technology from Xiangnan College, in 2021. She is currently pursuing the master's degree in electronic information with Hengyang Normal University, Hunan, China. Her current research interests include image encryption and image privacy protection.