Signal Processing 196 (2022) 108536

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Short communication

A novel image encryption algorithm with deep neural network

Chen Wang, Ye Zhang[*]

Department of Electronic Information Engineering, Nanchang University, Nanchang 330031, China

ARTICLE INFO

Article history:
Received 18 November 2021
Revised 1 March 2022
Accepted 5 March 2022
Available online 8 March 2022

MSC:
00-01
99-00

Keywords:
Image encryption
Deep neural network
Logistic map

ABSTRACT

A novel image encryption algorithm based on deep neural network (DNN) is proposed. First, a new encryption unit with the deep neural network (EDNN) is designed. The EDNN is a multilayer forward neural network, and the weight matrices of the EDNN are composed of some scrambled discrete cosine transform (DCT) coefficients matrices to encrypt directly the original image. Then, the original image can be recovered by a decryption unit with the deep neural network (DDNN). For the DDNN, its network structure is symmetric with the EDNN, and the original image can be recovered with two pursuit algorithms corresponding to two activation functions of the EDNN, respectively. The experimental results show the effectiveness of the EDNN for image encryption, and demonstrate that the proposed method can effectively resist brute force attack, statistical attack, differential attack and cutting attack.

1. Introduction

Image encryption is a fundamental task of image processing, and aims to guarantee the image security during transmission or storage. Generally, the image encryption schemes are based on the pixel permutation and the pixel diffusion [1–3]. Recently, the DNN has been applied into some image encryption schemes because of its excellent generalization performance and nonlinearity [4–10]. In [4], a convolutional neural network was used to generate plaintext related chaotic pointer to control the scrambling operation of two images, where the chaotic sequence was employed as convolution kernel of convolution neural network. In [5], the hyperchaotic system was combined with the deep convolutional generative adversarial networks to generate a random sequence as a key stream for scrambling and diffusion of the cipher image. In [6], a random matrix was generated by extracting a specific range of random numbers from the original image, and then the matrix was used to train the stacked auto encoders to generate a secure key for optimizing the performance of image encryption scheme. Ding et al. [7] utilized the generative adversarial network to construct a key generation network as a stream cipher generator to generate the private key. In the above-described image encryption approaches, the DNN is used as a pseudo-random number sequence generator to generate a key stream sequence. The key stream sequence is unpredictable, non-reproducible, and conforms to the requirements of an image encryption system. In [8], the initial values of network connection weights and biases of the multilayer perceptron were random numbers generated by using the logistic map. After training the multilayer perceptron, the biases and weights matrices were bitxor with the subimages to obtain encrypted subimages. Suhail and Sankar [9] used a shuffling matrix generated by the logistic map to scramble the original image, and then the scrambled image was compressed by an autoencoder. The compressed image was bitxor with a chaotic sequence to obtain the encrypted image. Duan et al. [10] used the random gradient method to train two variational auto-encoder generation models, and then two trained network parameters were divided. The divided network parameters were employed as the network parameters of the generated model to encrypt the original image. In the above-mentioned methods, the network parameters of the DNN are regarded as secret keys for image encryption, and the network parameters should be transmitted to the receiver. However, the network parameters of the DNN are large, which enlarge network transmission bandwidth and reduce transmission efficiency.

To enhance the security and reduce the consumption of keys of image encryption algorithm, we propose a novel encryption unit by DNN, i.e., EDNN, which is used as an image encryption generator to achieve image encryption directly. And the initial values and the system parameters of the logistic map are served as the secret keys rather than the network parameters of the EDNN. The EDNN is a multilayer forward neural network and some scrambled DCT coefficients matrices are employed as the network weight matrices. The scrambled DCT coefficients matrices are obtained by scrambling the row orders of some DCT coefficients matrices with the logistic map. The network structure of the decryption unit, i.e.,

[*] Corresponding author.
E-mail address: zhangye@ncu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.sigpro.2022.108536
0165-1684/© 2022 Elsevier B.V. All rights reserved.

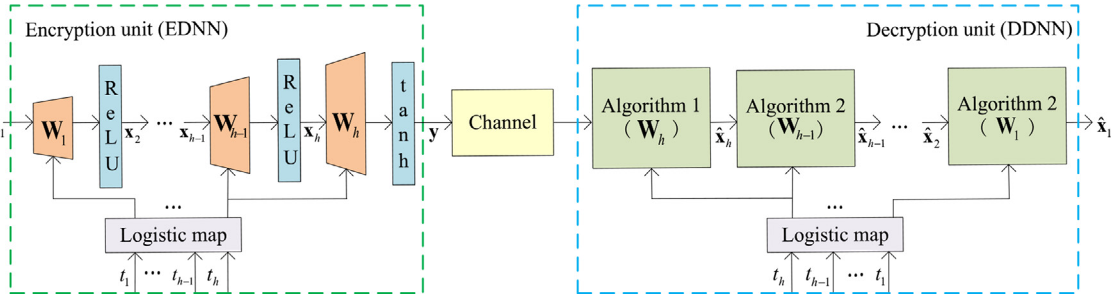


Fig. 1. The block diagram of proposed encryption and decryption method, and the network structure of the EDNN.

DDNN, is symmetric with the EDNN, and the original image can be recovered from the encryption image using two pursuit algorithms. The experimental results show that the proposed encryption system has the good performances of resisting exhaustive attack, statistical attack, differential attack and cutting attack.

2. The EDNN and the DDNN

The proposed encryption scheme includes two parts, i.e., encryption unit (EDNN) and decryption unit (DDNN), and the block diagram of the proposed scheme is shown in Fig. 1. The original image $\mathbf{x}_1$ is encrypted directly by the EDNN to obtain the encrypted image $\mathbf{y}$. In the DDNN, since the activation function of the output layer of the EDNN is the tanh function, we introduce the Algorithm 1 to estimate $\mathbf{x}_h$. The activation function of jth hidden layer of the EDNN is the ReLU function, then the Algorithm 2 can be utilized to estimate $\mathbf{x}_j$ $(j = h-1, h-2, \ldots, 1)$.

Algorithm 1 FISTA.

```
Input: $\hat{\mathbf{y}}$, $\mathbf{W}_h$, $\gamma$, the number of iterations $K$
Initialization: $\mathbf{d}(0) = \hat{\mathbf{x}}_h(0) = \mathbf{0}$, $t(0) = 1$
$\alpha = 1/p_{\max}(\mathbf{W}_h^{\mathrm{T}}\mathbf{W}_h)$, where $p_{\max}(\cdot)$ is to get the maximal value of the eigenvalues
for $k = 0 : K$ do
    $\mathbf{g} \leftarrow \mathbf{W}_h^{\mathrm{T}}\tanh'\big(\mathbf{W}_h\hat{\mathbf{x}}_h(k)\big) \circ \big[\tanh\big(\mathbf{W}_h\hat{\mathbf{x}}_h(k)\big) - \mathbf{y}\big]$, where $\circ$ is the element-wise product.
    $\mathbf{d}(k+1) \leftarrow \mathrm{ReLU}\big(\hat{\mathbf{x}}_h(k) - \alpha \cdot (\mathbf{g} + \gamma\mathbf{1})\big)$
    $t(k+1) \leftarrow \frac{1+\sqrt{1+4t(k)^2}}{2}$
    $\hat{\mathbf{x}}_h(k+1) \leftarrow \mathbf{d}(k+1) + \frac{t(k)-1}{t(k+1)}(\mathbf{d}(k+1) - \mathbf{d}(k))$
end for
Output: $\hat{\mathbf{x}}_h(K)$
```

The network structure of the EDNN is also shown in Fig. 1. The network structure of the EDNN consists of three parts: (1) the input layer, $\mathbf{x}_1 \in R^{n_1 \times 1}$ is a vector of an original image; (2) $h-1$ hidden layers, $\mathbf{x}_{i+1} \in R^{n_{i+1} \times 1}$ is the output of ith hidden layer and expressed as

$$\mathbf{x}_{i+1} = \mathrm{ReLU}(\mathbf{W}_i\mathbf{x}_i), \quad \text{for} \quad i = 1, 2, \ldots, h-1 \tag{1}$$

where $\mathbf{x}_i$ is the input of ith hidden layer, $\mathbf{W}_i \in R^{n_{i+1} \times n_i}$ is the ith layer network weight matrix, $n_{i+1} \gg n_i$. $\mathrm{ReLU}(u) = \max(0, u)$ is an activation function, i.e., if $\mathbf{W}_i\mathbf{x}_i \le 0$, $\mathbf{x}_{i+1} = 0$, $\mathbf{x}_{i+1}$ is sparse; (3) the output layer, $\mathbf{y} \in R^{n_{h+1} \times 1}$ is a vector of the encryption image,

$$\mathbf{y} = \tanh(\mathbf{W}_h\mathbf{x}_h) \tag{2}$$

where $\mathbf{W}_h \in R^{n_{h+1} \times n_h}$ is a network weight matrix of the output layer, $n_{h+1} > n_h$, and $\tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$ is a nonlinear activation function, then the EDNN is a nonlinear encryption network.

For the EDNN, the network weight matrices are comprised of some scrambled DCT coefficients matrices, the orders of the rows of the DCT coefficients matrices are scrambled with the logistic map. To obtain the ith layer network weight matrix $\mathbf{W}_i \in R^{n_{i+1} \times n_i}$ $(i = 1, 2, \ldots, h)$, a DCT coefficients matrix $\bar{\mathbf{W}}_i \in R^{n_{i+1} \times n_i}$ is scrambled with the logistic map. The logistic map for $\bar{\mathbf{W}}_i$ is defined as

$$r_{i,d+1} = \mu_i r_{i,d}(1 - r_{i,d}), r_{i,d} \in (0, 1) \tag{3}$$

where $r_{i,d}$ is the value for d iterations, and μ_i is the system parameter of the logistic map with the range of [3.57, 4]. The random sequence $\mathbf{P}_i$ of length n_{i+1} is generated with the logistic map under the initial value $r_{i,0}$ and the parameter μ_i. Then, the orders of the rows of the matrix $\bar{\mathbf{W}}_i$ are sorted in ascending order of $\mathbf{P}_i$ to obtain $\mathbf{W}_i$. The EDNN can be used as an image encryption generator to encrypt the original image directly.

Algorithm 2 AD-LPMM.

```
Input: $\hat{\mathbf{x}}_{j+1}$, $\mathbf{W}_j$, $\lambda_j$, $\rho_j$, the number of iterations $K_j$
Initialization: $\mathbf{u}(0) = \mathbf{b}(0) = \mathbf{0}$, $\hat{\mathbf{x}}_j(0) = \mathbf{0}$
$\mathbf{S}$ = find($\hat{\mathbf{x}}_{j+1}$), where find($\hat{\mathbf{x}}_{j+1}$) function is to find the index of non-zero elements in $\hat{\mathbf{x}}_{j+1}$. $\mathbf{S^c}$ is the complementary set of $\mathbf{S}$
$\mathbf{W}_j^{\mathbf{S}} = \mathbf{W}_j(\mathbf{S}, :)$, $\mathbf{W}_j^{\mathbf{S^c}} = \mathbf{W}_j(\mathbf{S^c}, :)$
$\eta = p_{\max}\Big(\mathbf{W}_j^{\mathbf{S}^{\mathrm{T}}}\mathbf{W}_j^{\mathbf{S}} + \rho_j\mathbf{W}_j^{\mathbf{S^c}\mathrm{T}}\mathbf{W}_j^{\mathbf{S^c}}\Big) + 0.1$
for $k = 0 : K_j$ do
    $\hat{\mathbf{x}}_j(k+1) \leftarrow \mathrm{ReLU}\Big(\hat{\mathbf{x}}_j(k) - \frac{1}{\eta}\mathbf{W}_j^{\mathbf{S}^{\mathrm{T}}}\Big(\mathbf{W}_j^{\mathbf{S}}\hat{\mathbf{x}}_j(k) - \hat{\mathbf{x}}_{j+1}^{\mathbf{S}}\Big)$
        $-\frac{\rho_j}{\eta}\mathbf{W}_j^{\mathbf{S^c}\mathrm{T}}\Big(\mathbf{W}_j^{\mathbf{S^c}}\hat{\mathbf{x}}_j(k) - \mathbf{b}(k) - \mathbf{z}(k)\Big) + \frac{\lambda_j}{\eta}\mathbf{1}\Big)$
    $\mathbf{b}(k+1) \leftarrow -\mathrm{ReLU}\Big(\mathbf{u}(k) - \mathbf{W}_j^{\mathbf{S^c}}\hat{\mathbf{x}}_j(k+1)\Big)$
    $\mathbf{u}(k+1) \leftarrow \mathbf{u}(k) + \mathbf{b}(k+1) - \mathbf{W}_j^{\mathbf{S^c}}\hat{\mathbf{x}}_j(k+1)$
end for
Output: $\hat{\mathbf{x}}_j(K_j)$
```

For the DDNN, the network structure of the DDNN is symmetric with the EDNN. The activation function of the output layer of the EDNN is the tanh function, $\mathbf{x}_h$ is sparse and nonnegative, the following optimization problem can be employed to estimate $\mathbf{x}_h$

$$\hat{\mathbf{x}}_h = \arg\min_{\hat{\mathbf{x}}_h} \frac{1}{2}||\mathbf{y} - \tanh(\mathbf{W}_h\hat{\mathbf{x}}_h)||_2^2 + \gamma\mathbf{1}^{\mathrm{T}}\hat{\mathbf{x}}_h, \ \text{s.t. } \hat{\mathbf{x}}_h \ge \mathbf{0} \tag{4}$$

where $\|\cdot\|_2^2$ represents the square of ℓ_2 norm. $\gamma > 0$ is a Lagrangian constant. $\mathbf{1} = [1, 1, \ldots, 1]^{\mathrm{T}} \in R^{n_h \times 1}$, and T represents the transpose operation. The above optimization problem can be solved with the fast iterative shrinkage-thresholding algorithm (FISTA) [11], as described in Algorithm 1.

The activation function of jth hidden layer of the EDNN is the ReLU function, $\mathbf{x}_j$ is sparse and nonnegative, then $\mathbf{x}_j$ can be estimated by the following optimization problem

$$\hat{\mathbf{x}}_j = \arg\min_{\hat{\mathbf{x}}_j} \frac{1}{2}\left\|\hat{\mathbf{x}}_{j+1}^{\mathbf{S}} - \mathbf{W}_j^{\mathbf{S}}\hat{\mathbf{x}}_j\right\|_2^2 + \lambda_j\mathbf{1}^{\mathrm{T}}\hat{\mathbf{x}}_j, \ \text{s.t. } \hat{\mathbf{x}}_j \ge \mathbf{0}, \ \mathbf{W}_j^{\mathbf{S^c}}\hat{\mathbf{x}}_j = \mathbf{0} \tag{5}$$

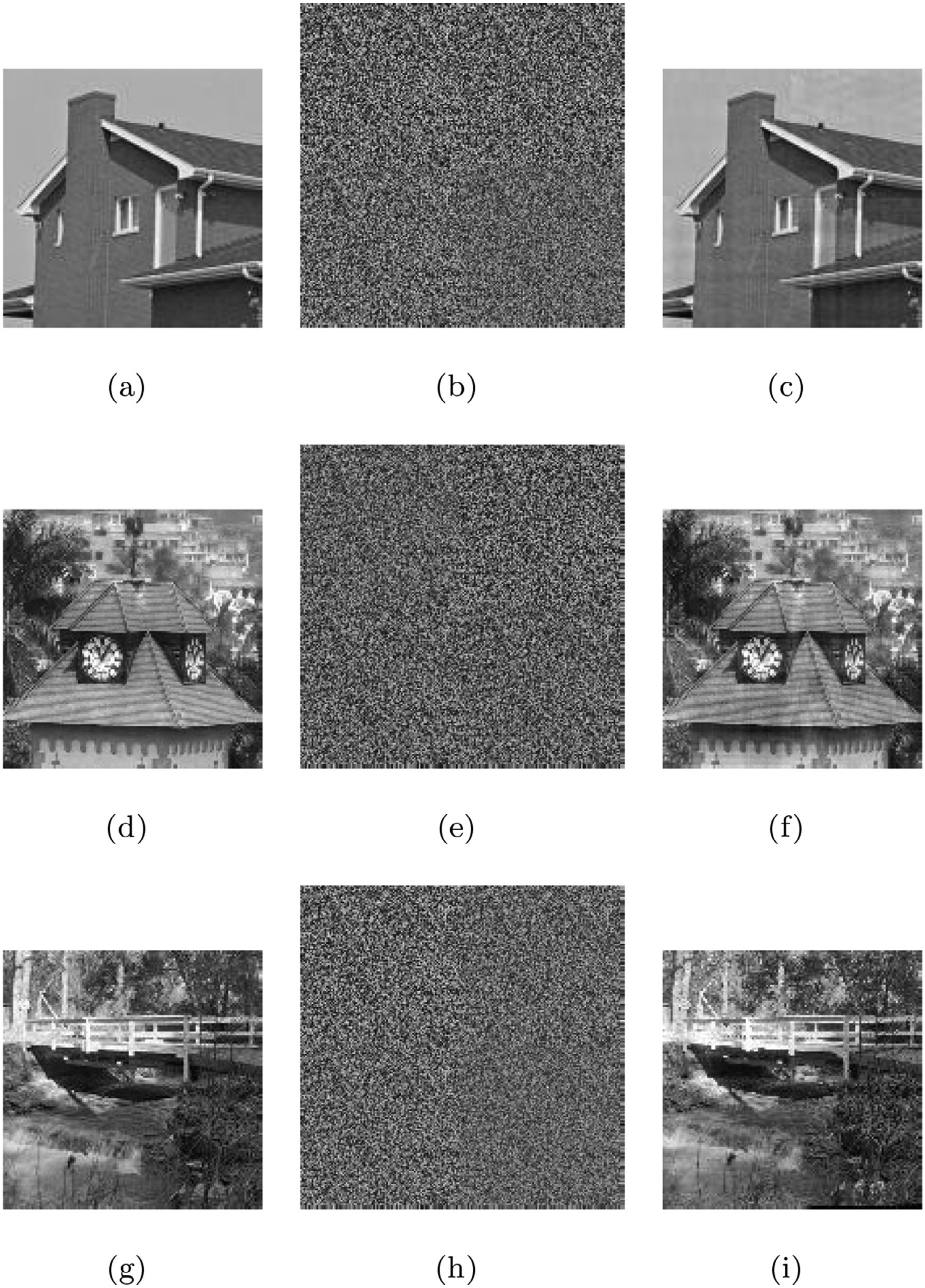


Fig. 2. Encryption and decryption results. (a) The original image “House”; (b) The encryption “House”; (c) The decryption “House”, PSNR = 38.2129dB; (d) The original image “Pagodas”; (e) The encryption “Pagodas”; (f) The decryption “Pagodas”, PSNR = 30.9388dB; (g) The original image “Bridge”; (h) The encryption “Bridge”; (i) The decryption “Bridge”, PSNR = 33.3939dB.

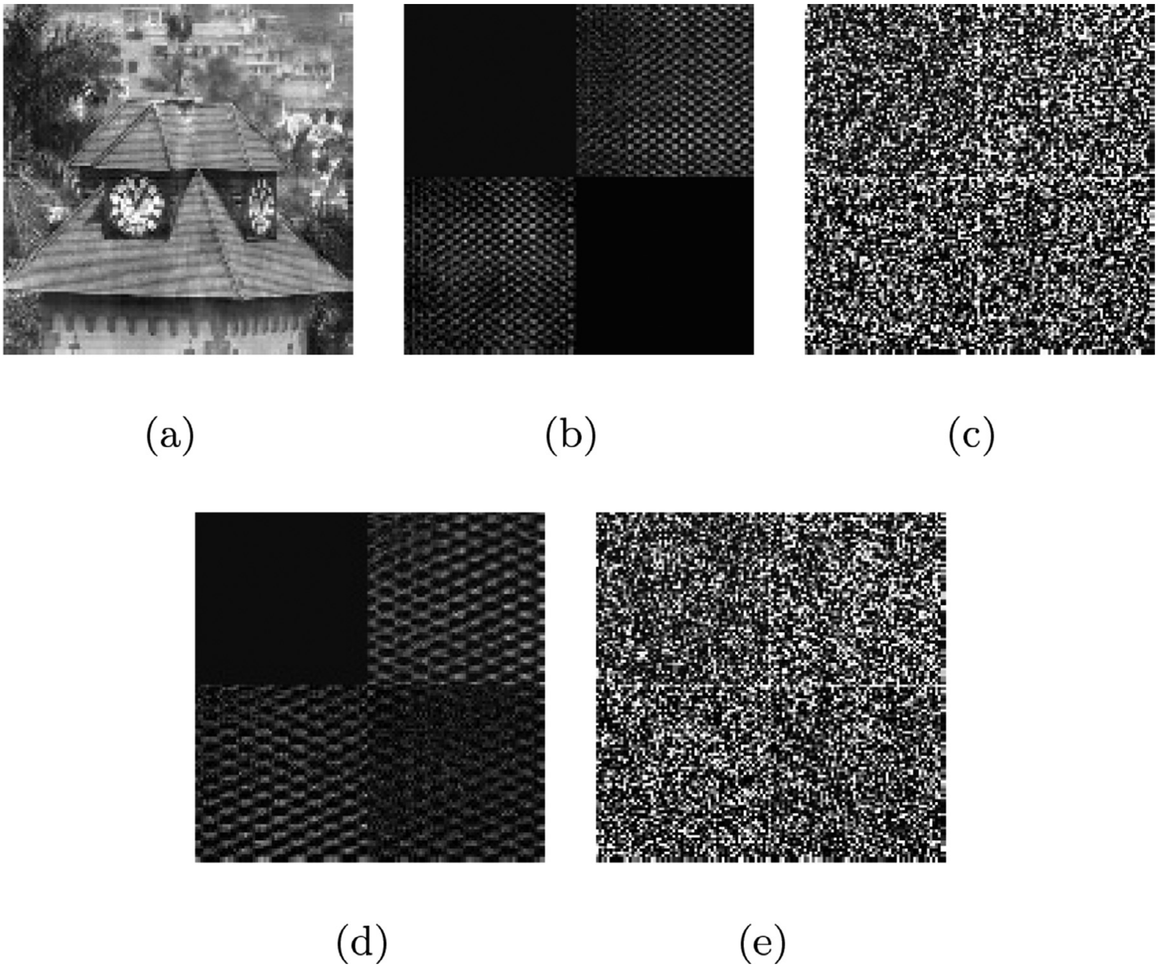


Fig. 3. Decryption image of “Pagodas” with adjusted keys. (a) correct secret keys; (b) $r_{1,0} + \Delta = 3.99 + 10^{-15}$; (c) $r_{2,0} + \Delta = 3.99 + 10^{-15}$; (d) $t_{1,0} + \Delta = 0.3 + 10^{-15}$; (e) $t_{2,0} + \Delta = 0.4 + 10^{-15}$.

where $\hat{\mathbf{x}}_j$ is the estimation of the output of the jth layer, and $\hat{\mathbf{x}}_{j+1}$ is the input of the jth layer in the DDNN. $\hat{\mathbf{x}}^{\mathbf{S}}_{j+1}$ is a subset of all nonzero elements of $\hat{\mathbf{x}}_{j+1}$ and $\mathbf{S}$ is a set consisting of the index of the nonzero elements, $\mathbf{S^c}$ is the complementary set of $\mathbf{S}$. $\mathbf{W}^{\mathbf{S}}_j$ is a sub-matrix of $\mathbf{W}_j$, which contains the rows from $\mathbf{W}_j$ indexed by $\mathbf{S}$, i.e., $\mathbf{W}^{\mathbf{S}}_j \hat{\mathbf{x}}_j \neq \mathbf{0}$.

To solve the above optimization problem, we define an auxiliary variable $\mathbf{b} = \mathbf{W}^{\mathbf{S^c}}_j \hat{\mathbf{x}}_j$, Eq. (5) can be rewritted as

$$\begin{aligned} &\min_{\hat{\mathbf{x}}_j,\mathbf{b},\mathbf{u}} \tfrac{1}{2}\left\|\hat{\mathbf{x}}^{\mathbf{S}}_{j+1} - \mathbf{W}^{\mathbf{S}}_j \hat{\mathbf{x}}_j\right\|_2^2 + \lambda_j \mathbf{1}^{\mathrm{T}} \hat{\mathbf{x}}_j + \tfrac{\rho_j}{2}\left\|\mathbf{b} - \mathbf{W}^{\mathbf{S^c}}_j \hat{\mathbf{x}}_j + \mathbf{u}\right\|_2^2, \\ &\quad \text{s.t. } \hat{\mathbf{x}}_j \geq \mathbf{0}, \ \ \mathbf{b} = \mathbf{0} \end{aligned} \tag{6}$$

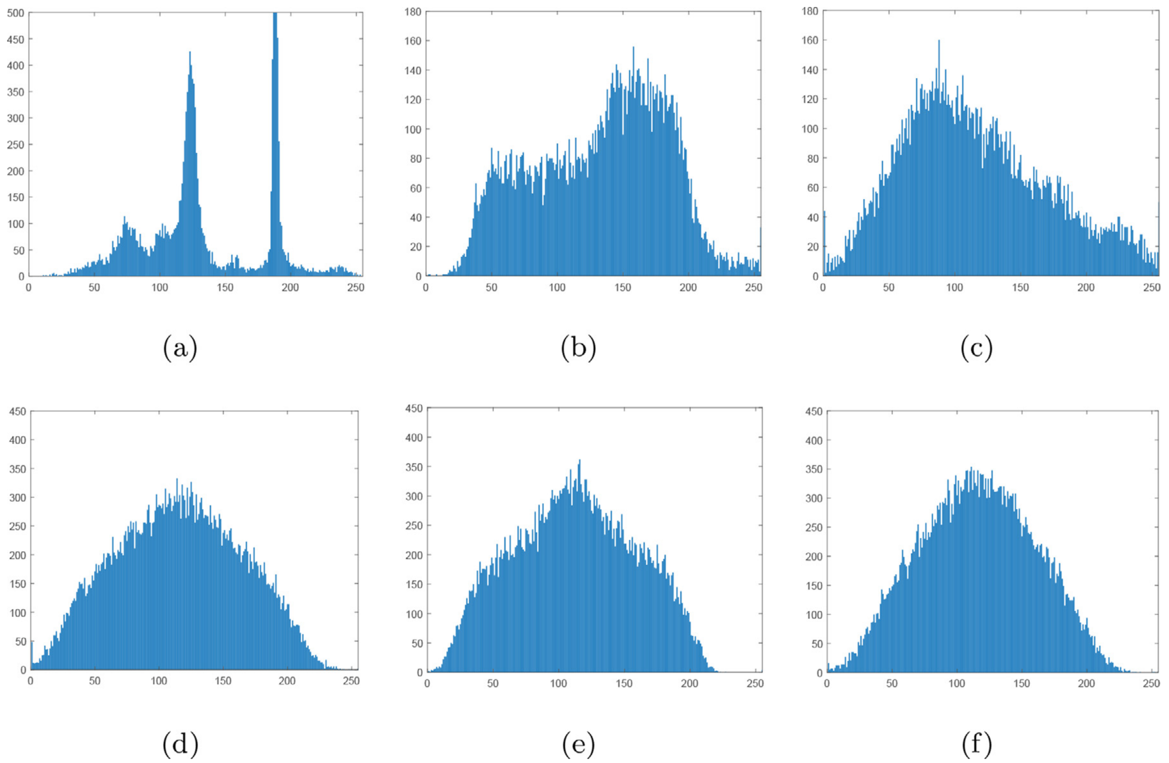
where λ_j and ρ_j are the Lagrangian constants, and $\mathbf{u}$ is an intermediate variable during iterations. $\hat{\mathbf{x}}_j$ can be estimated with the alternating direction linearized proximal method of multipliers (AD-LPMM) [12] algorithm, as described in Algorithm 2. Finally, the estimation of the original image $\mathbf{x}_1$ can be obtained.

3. Simulation experiment and performance analysis

To verify the security and effectiveness of the EDNN for image encryption, the original images “House”, “Pagodas”, “Bridge”, “Mandrill” and “Peppers” of size 128 × 128 are used as test images. The original images “House”, “Pagodas” and “Bridge” are shown in Fig. 2(a), (d) and (g), respectively. The original image is divided into four nonoverlapping sub-blocks of the same size 64 × 64. Each sub-block is converted into a vector of the size 4096 × 1, and then the vector is normalized to the range of 0 to 1, which is used as the input of the EDNN. The EDNN is composed of three layers. In these experiments, $h = 2$, $n_1 = 4096$, $n_2 = 9025$, and $n_3 = 10000$. The system parameters and the initial values of the logistic map are set as $r_{1,0} = 0.3$, $r_{2,0} = 0.4$, $\mu_1 = \mu_2 = 3.99$. For the DDNN, $K = 500$, $K_1 = 5000$, $\lambda_1 = 0.01$, $\rho_1 = 10^{-4}$, $\gamma = 10^{-5}$. The encryption images “House”, “Pagodas” and “Bridge” of size 200 × 200 are shown in Fig. 2(b), (e) and (h), respectively. Any meaningful information cannot be obtained from the encryption images. The decryption images “House”, “Pagodas” and “Bridge” are shown in Fig. 2(c), (f) and (i), respectively. The quality of decryption images is evaluated by peak-to-peak signal-to-noise ratio (PSNR). The PSNR values of the decryption images are 38.2129dB, 30.9388dB and 33.3939dB, respectively. As can be seen, the decryption images have good reconstruction quality.

In the proposed cryptosystem, the given initial values ($r_{1,0}$, $r_{2,0}$) and the system parameters (μ_1, μ_2) of the logistic map are the secret keys. If the operation accuracy of the computer is 10^{15}, the key space can be calculated as $(10^{15})^4 = 10^{60} \approx 2^{199}$. It is larger than the theoretical security requirement of the key space 2^{100}. If the number of network layers is increasing, the key space will also be enlarged. The key space is large enough to resist exhaustive attacks.

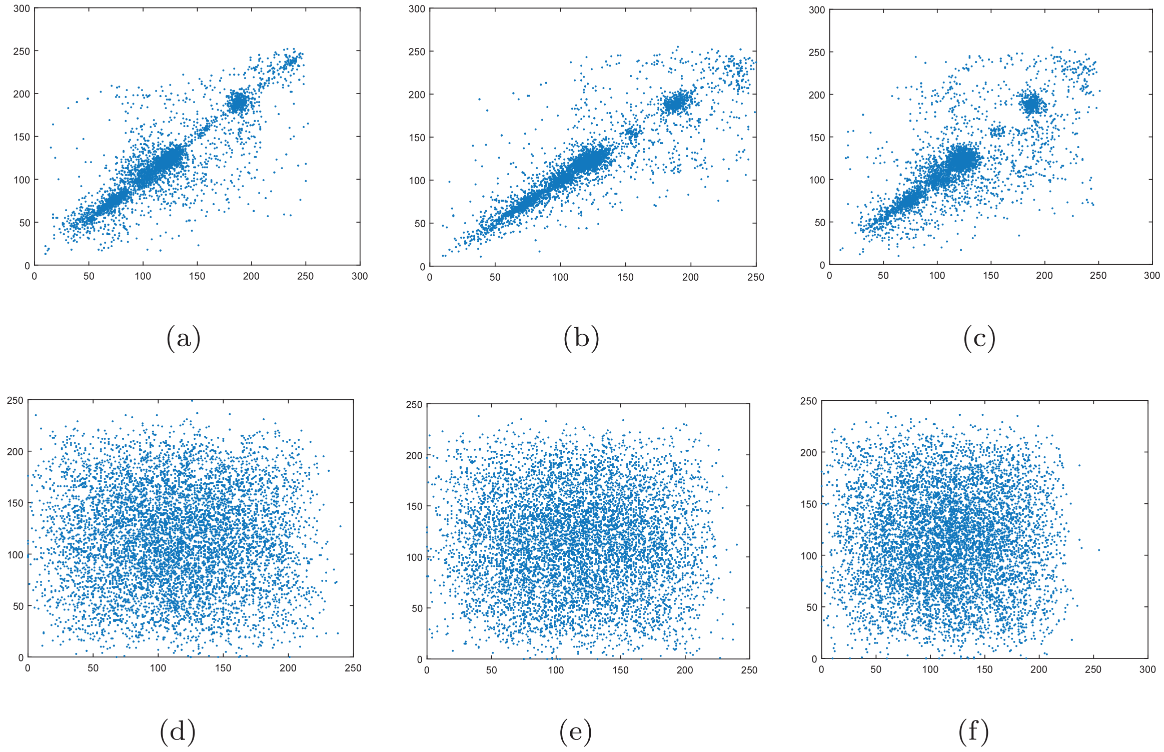
To verify key sensitivity, the key ($r_{1,0}$, $r_{2,0}$, $t_{1,0}$ or $t_{2,0}$) is changed slightly change with $\Delta = 10^{-15}$ during decryption process. The experimental results are shown in Fig. 3. It is obvious that any useful information cannot be obtained from the decryption image



(a) (b) (c)

(d) (e) (f)

Fig. 4. Histograms of original images and encryption images. (a) The original image of “House”; (b) The original image of “Pagodas”; (c) The original image of “Bridge”; (d) The encryption image of “House”; (e) The encryption image of “Pagodas”; (f) The encryption image of “Bridge”.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Correlation between adjacent pixels before and after encryption. (a) The horizontal direction of House’s original image; (b) The vertical direction of House’s original image; (c) The diagonal direction of House’s original image; (d) The horizontal direction of House’s encryption image; (e) The vertical direction of House’s encryption image; (f) The diagonal direction of House’s encryption image.

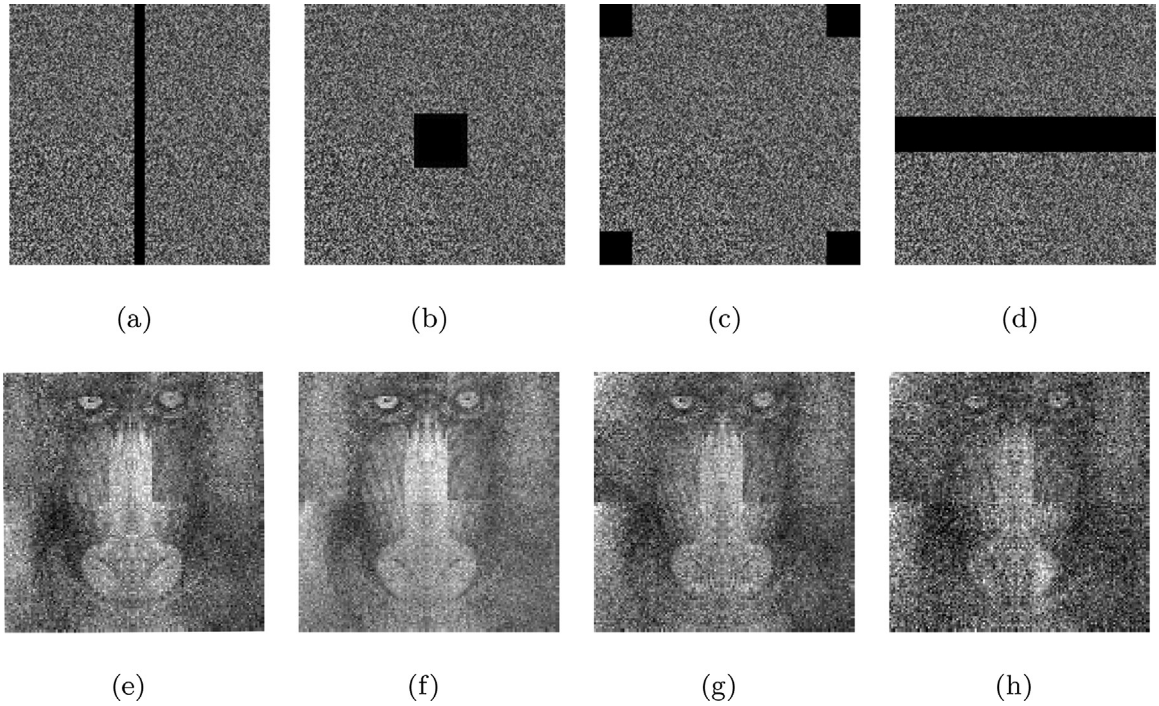


Fig. 6. (a), (b), (c), (d) Encryption images with 5%, 10%, 15%, 20% occlusion in different positions, respectively. (e), (f), (g), (h) are the corresponding decryption images "Mandrill".

even if the key has a slight change. The proposed image encryption algorithm is highly sensitive to the initial keys and can resist the statistical analysis attack.

To resist the impact of statistical attacks, it is necessary to break the statistical similarity of original images in encryption images. Fig. 4 shows the histograms of the original images "House", "Pagodas" and "Bridge" and the corresponding encryption images, respectively. It can be found that the histogram distributions of the encryption images are similar to each other, even though the histogram distributions of these original images are quite different. The proposed method can resist statistical attacks since the histograms of different encryption images are similar.

A good encryption system should break the correlation between adjacent pixels in original images. 8000 pairs of adjacent pixels in the horizontal, vertical, and diagonal directions from the original images "House", "Pagodas", "Bridge" and the corresponding encrypted images are selected randomly, and the correlation coefficient between adjacent pixels is calculated, respectively. The correlation diagrams among adjacent pixels in horizontal, vertical and diagonal directions in image "House" are shown in Fig. 5. It is obvious that the correlation of the encryption image is greatly weaker than that of the original image. Therefore, it means that the proposed image encryption algorithm based on the EDNN reduces the correlation among adjacent pixels of the encryption image. The correlation coefficients of the encryption images with the proposed image encryption algorithm, and these of Refs. [2,10] are listed in Table 1, which indicate that the proposed image encryption algorithm has a better performance. For the proposed encryption method, the correlation coefficient of the original images are close to 1, while the correlation coefficient of the encryption images are small even almost close to 0. It is showed that the proposed encryption method has better encryption effect and anti-statistical attack performance.

Differential attack is a significant method to test the plaintext sensitivity of the algorithm. The number of pixel change rate (NPCR) and the unified average changing intensity (UACI) are common indicators to evaluate the impact of differential attack [3].

Table 1
Correlation coefficients for adjacent pixels.

Algorithm	Image	Horizontal	Vertical	Diagonal
	Original House	0.9092	0.9403	0.9403
Our algorithm	Encrypted House	0.0016	−0.0019	0.0012
Ref. [2]	Encrypted House	−0.0095	−0.0108	0.0023
Ref. [10]	Encrypted House	0.0354	0.0193	0.0255
	Original Pagodas	0.8094	0.8379	0.7543
Our algorithm	Encrypted Pagodas	−0.0017	0.0037	−0.0046
Ref. [2]	Encrypted Pagodas	0.0075	−0.0071	−0.0063
Ref. [10]	Encrypted Pagodas	−0.0120	0.0150	0.0217
	Original Bridge	0.7995	0.8348	0.7619
Our algorithm	Encrypted Bridge	−0.0049	−0.0052	−0.0035
Ref. [2]	Encrypted Bridge	−0.0069	−0.0056	−0.0160
Ref. [10]	Encrypted Bridge	−0.0101	0.0166	−0.0111

Table 2
NPCR and UACI values of the cipher images.

Algorithm	Image	NPCR	UACI
Our algorithm	Bridge	99.9800%	33.4365%
	Mandrill	99.9875%	33.4600%
	Peppers	99.9900%	33.4589%
Ref. [2]	Bridge	99.5728%	33.3983%
	Mandrill	99.5728%	33.4294%
	Peppers	99.6277%	33.5189%

The NPCR and the UACI values of the test images "Bridge", "Mandrill" and "Peppers" are shown in Table 2. It is showed that the NPCR values of the proposed image encryption algorithm are larger than these values of the method [2], and the UACI values of the proposed image encryption algorithm are close to the theoretical value. Therefore, the proposed image encryption algorithm can resist differential attack.

The encrypted image may be intercepted by the attacker in the process of transmission. Fig. 6 shows the corresponding decryption images of the encryption images "Mandrill" with different cutting ratios and different cutting regions, respectively. As shown in Fig. 6, although the decryption images become fuzzier with the increase of occlusion size, the main information of the original image remains recognizable. Therefore, the image encryption algorithm based on the EDNN has a certain degree of robustness against cutting attack.

4. Conclusion

A novel encryption unit based on the DNN for image encryption is proposed. In the encryption unit, the DNN that does not need to be trained can be applied to encrypt directly the original image. In the decryption unit, the original image is recovered from the encryption image with two pursuit algorithms. The experiment results show that the proposed method is easily implementable, secure and effective, and it has good performances of standing against exhaustive attack, statistical attack, differential attack and cutting attack.

Declaration of Competing Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

CRediT authorship contribution statement

Chen Wang: Data curation, Writing – original draft, Software. **Ye Zhang:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61866024).

References

[1] X.Y. Wang, M.C. Zhao, An image encryption algorithm based on hyperchaotic system and DNA coding, Opt. Laser Technol. 143 (2021) 107316, doi:10.1016/j.optlastec.2021.107316.

[2] K.U. Shahna, A. Mohamed, A novel image encryption scheme using both pixel level and bit level permutation with chaotic map, Appl. Soft Comput. 90 (2020) 106162, doi:10.1016/j.asoc.2020.106162.

[3] H.S. Ye, N.R. Zhou, L.H. Gong, Multi-image compression-encryption scheme based on quaternion discrete fractional Hartley transform and improved pixel adaptive diffusion, Signal Process. 175 (2020) 107652, doi:10.1016/j.sigpro.2020.107652.

[4] Z.L. Man, J.Q. Li, et al., Double image encryption algorithm based on neural network and chaos, Chaos Solitons Fractals. 152 (2021) 111318, doi:10.1016/j.chaos.2021.111318.

[5] P. Fang, H. Liu, C.M. Wu, A novel chaotic block image encryption algorithm based on deep convolutional generative adversarial networks, IEEE Access. 9 (2021) 18497–78517, doi:10.1109/access.2020.3040573.

[6] S.R. Maniyath, V. Thanikaiselvan, An efficient image encryption using deep neural network and chaotic map, Microprocessors Microsyst. 77 (2020) 103134, doi:10.1016/j.micpro.2020.103134.

[7] Y. Ding, F.Y. Tan, et al., DeepKeyGen: a deep learning-based stream cipher generator for medical image encryption and decryption, IEEE Trans. Neural Netw. Learn. Syst. (2021) 1–15., doi:10.1109/TNNLS.2021.3062754.

[8] A.N.K. Telem, C.M. Segning, et al., A simple and robust gray image encryption scheme using chaotic logistic map and artificial neural network, Adv. Multimedia 2014 (2014) 602921, doi:10.1155/2014/602921.

[9] K.A. Suhail, S. Sankar, Image compression and encryption combining autoencoder and chaotic logistic map, Iranian J. Sci. Technol. Trans. A-Sci. 44 (4) (2020) 1091–1100, doi:10.1007/s40995-020-00905-4.

[10] X.T. Duan, J.J. Liu, E. Zhang, Efficient image encryption and compression based on a VAE generative model, J. Real-Time Image Process. 16 (3) (2019) 765–773, doi:10.1007/s11554-018-0826-4.

[11] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202, doi:10.1137/080716542.

[12] A. Beck, First-Order Methods in Optimization, Society for Industrial and Applied Mathematics, Philadelphia, 2017.