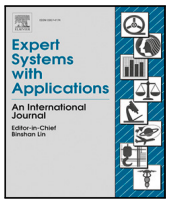
Expert Systems With Applications 279 (2025) 127404

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Semantically enhanced selective image encryption scheme with parallel computing

Buyu Liu [a], Wei Song [a,*], Mingyi Zheng [a], Chong Fu [a,b], Junxin Chen [c], Xingwei Wang [a]

[a] School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
[b] Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, China
[c] School of Software, Dalian University of Technology, Dalian 116621, China

ARTICLE INFO

Keywords:
Image ROI encryption
Salient object detection
Chaos
Simultaneous permutation–diffusion
Parallel computing

ABSTRACT

Recently, an increasing number of ROI (regions of interest) encryption algorithms have been proposed to efficiently encrypt the sensitive regions of image. Due to the powerful feature extraction capabilities of deep learning (DP), many DP-based object detection models have been increasingly applied to ROI encryption. However, some models with a large number of parameters are inefficient and not suitable for real-time detection, and the detected ROI often include some redundant regions. Moreover, the following encryption operations are always in serial manner, leaving room for improvement. To address these issues, we present a semantically enhanced selective image encryption scheme with parallel computing. The deep salient object detection (SOD) model is first lightweighted to improve detection efficiency. Then, the sensitive region is cropped based on the boundary information from the output saliency map, resulting in an ROI that removes redundant regions without revealing sensitive object information. In encryption stage, the three color channels of each pixel are assigned to a group and encrypted in parallel to further improve the efficiency. Furthermore, to enhance the practicality, we embedded the side information of the ROI into the image, eliminating the need to separately distribute the image and the corresponding side information. Finally, we carry out security and efficiency analyses, and the results demonstrate that the proposed encryption scheme can enable efficient and secure detection of sensitive regions, along with corresponding encryption protection.

1. Introduction

With the rapid development of the Internet, people are sharing various images on the internet every day. However, without necessary protection, hackers can easily obtain people's private information and sell them to criminals. In order to guarantee information security, many image cryptosystems have been proposed.

With image generation/capture devices becoming increasingly powerful and the increase in network bandwidth, the use of large-capacity and high-resolution images has become increasingly frequent. Conventional encryption algorithms such as DES, AES, RSA and some existing image encryption schemes (Hu et al., 2024; Kocak et al., 2024; Lai et al., 2022; Peng et al., 2024; Song et al., 2023; Song, Fu, Zheng, et al., 2024; Zhang et al., 2023; Zhou et al., 2024) can enable the protection of image data, but have some operations on non-sensitive regions. The main reason is that many methods do not take image semantic features into account and often encrypt the entire image, leading to reduced efficiency.

As a matter of fact, not all regions in image are equally sensitive, therefore, digital images can be protected by encrypting sensitive regions which are called Regions of Interest (ROI). Table 1 lists some recent ROI detection methods. In Singh, Devi, et al. (2022), ROI were detected manually and in Balasamy and Suganyadevi (2021), Cai et al. (2024), Kiran and Parameshachari (2022), Kiran et al. (2020), Liu, Zhang, et al. (2020), Murali et al. (2023), Ramacharya et al. (2023), Su et al. (2023), some machine learning methods are exploited to detect the sensitive regions of the images. However, compared with the increasingly developed deep learning-based feature attraction approaches (Zhang et al., 2025; Zou et al., 2025), these conventional ROI detection schemes tend to have big limitations in speed and accuracy. Singh et al. (2024), Wang et al. (2023), Wang, Liu, et al. (2022), Wang et al. (2024) exploited instance segmentation to accurately locate sensitive regions of the image, yet they ignored the hiding of edge information, thus causing the risk of information leakage. Due to the validity of YOLO in object detection, (Asgari-Chenaghlu et al., 2021;

* Corresponding author.
E-mail addresses: 2301807@stu.neu.edu.cn (B. Liu), songwei@mail.neu.edu.cn (W. Song), 2401802@stu.neu.edu.cn (M. Zheng), fuchong@mail.neu.edu.cn (C. Fu), junxinchen@ieee.org (J. Chen), wangxw@mail.neu.edu.cn (X. Wang).

https://doi.org/10.1016/j.eswa.2025.127404
Received 13 December 2024; Received in revised form 13 March 2025; Accepted 22 March 2025
Available online 4 April 2025

Table 1
Several ROI detection methods and their drawbacks.

Refs.	ROI detection methods	Limitations
Singh, Devi, et al. (2022)	Manual segmentation	Lack of real-time considerations.
Liu, Zhang, et al. (2020)	HOG + SVM	
Murali et al. (2023)	Orthogonal polynomial transformation	Limited feature extraction capabilities.
Su et al. (2023)	Multilayer cellular automata saliency detection	
Ramacharya et al. (2023) Balasamy and Suganyadevi (2021)	Fuzzy rules	
Cai et al. (2024)	KOS + VOS	
Kiran et al. (2020)	Active contour model	
Kiran and Parameshachari (2022)	Laplacian edge detection operator	
Wang, Liu, et al. (2022)	PSPNet + Faster-RCNN	Redundant category categorization.
Singh et al. (2024)	UNet3+	Edge information leakage
Wang et al. (2023, 2024)	Mask RCNN	Redundant category categorization and edge information leakage.
Chen and Yang (2022)	YOLOv3 + UNet	
Singh, Singh, et al. (2022) Sheela and Suresh (2024) Asgari-Chenaghlu et al. (2021) Priyanka et al. (2024) Zou et al. (2025) Wang et al. (2025)	YOLO series of algorithms	Redundant category categorization and large ROI.
Song et al. (2022)	Modified YOLOv4	

Priyanka et al., 2024; Sheela & Suresh, 2024; Singh, Singh, et al., 2022; Wang et al., 2025; Zou et al., 2025) employed different versions of YOLO without optimization to detect the ROI, and Song et al. (2022) presented a modified YOLOv4 (Bochkovskiy et al., 2020) to guarantee that the bounding box can cover all regions of the target. However, whether it is instance segmentation or object detection, they classify targets when performing ROI detection, which is unnecessary in image encryption. Furthermore, the detected ROI in the above methods tend to be a large rectangular box, where many redundant windows are concluded, thus affecting the efficiency of the algorithm. Besides that, all of the above methods use a serial encryption approach during the encryption stage, which results in limited encryption efficiency. Moreover, most methods directly transmit side information related to the ROI without adequate protection, and also increase the cost of data distribution.

In summary, we identify the following issues in existing ROI methods that still need to be addressed.

- The object detection stage should enhance protection for boundary regions while minimizing the inclusion of non-sensitive regions.
- The efficiency of both object detection and encryption stages should be improved to enable real-time protection of sensitive regions.
- ROI side information lacks protection, and transmitting it separately increases the distribution burden.

Having an intention to solve above problems, we present a novel and efficient algorithm for protecting image ROI. In object detection stage, a novel Patch-based Salient Region Detection (PSRD) method is introduced to optimize the detected sensitive regions. Here, we use deep-learning based salient object detection (SOD) to identify regions instead of object segmentation and detection, as it detects salient regions without categorizing the objects. It should be noted that most existing deep-SOD models (Pang et al., 2020; Qin et al., 2020, 2019; Wu et al., 2019; Zhao et al., 2020) tend to choose VGGs (Simonyan & Zisserman, 2014) or ResNets (He et al., 2015) as backbone, which contains convolutional layers that have enormous parameters, thus making these networks hard to deployed on device that has limited memory and computation resources. To improve the efficiency and practicality, we employ MobileNetV2-based (Sandler et al., 2018) lightweight Extremely-Downsampled Network (EDN-lite) (Wu, Liu, Zhang, et al., 2022) to efficiently detect the salient regions. The reason is that lightweighted EDN pays attention on low/high-level features learning at the same time, and it has higher detection accuracy and compared with some other methods while having fewer parameters. Specifically, the detected salient map is divided into multiple patches averagely. If there is a salient region in one patch, then the region is covered by a bounding box, which is determined by the boundary of detected salient region. In this way, we reduce the redundant data to be encrypted and the sensitive information in edge regions is well-protected in the subsequent encryption process.

In the encryption stage, the intrinsic properties of chaotic system, such as high sensitivity to initial conditions and good pseudo-randomness, make it well suited to cryptographic algorithms (Hua et al., 2020, 2021; Zhou et al., 2023). Here, we present a parallel chaos-based encryption method for fast encryption. For each pixel, its three 8-bit sub-pixel is randomly assigned to a new sequence, which is determined by a logistic map, and after assignment, the three pixel sequences are encrypted in parallel by an improved multi-dimensional hyperchaotic system (Wei & Li, 2022).

To ensure comprehensive protection and avoid the hassle of additional distribution, we aim to transmit all confidential data in a single transfer. Therefore, we employ reversible steganography to embed the ROI information into the ciphered image. Here, a reversible data hiding scheme based on the images texture (Jia et al., 2019) is used to hide the side information of the ROI inside the encrypted image. Here, ROI side information contains the number and the coordinate information of the ROI in each patch. Experimental results and security analyses demonstrate that our scheme outperforms state-of-the-art encryption algorithms on encryption efficiency and practicability.

Compared with existing schemes, the contributions of our scheme are summarized as follows,

- In ROI encryption, the proposed PSRD first integrates lightweight ROI detection with patch-level optimization, improving detection efficiency.
- The three color channels of ROI are encrypted in parallel, thus improving the efficiency.
- Steganographic protection of ROI side information enhances the security and practicality.

The rest of this article is structured as follows. In Section 2, we introduce deep learning-based SOD, LICC hyperchaotic system and a reversible data hiding method. Our ROI detection scheme and the parallel image encryption algorithm is presented in Section 3. Experimental results and corresponding security analyses are reported in Section 4. And Section 5 is the conclusion and some discussion about the future work.

2. Related work

2.1. Deep learning-based SOD

SOD is a means used by Computer Vision (CV) to mimic the human visual system to identify important regions in a target image (Qin et al., 2019). With the wide application of Convolutional Neural Network (CNN) (Lecun et al., 1998) and Vision Transformer (ViT) (Dosovitskiy et al., 2020) in the field of computer vision, the deep learning-based SOD algorithms (Pang et al., 2020; Qin et al., 2020, 2019; Wu et al., 2019; Wu, Liu, Zhang, et al., 2022; Zhao et al., 2020) can distinguish the salient regions in the image and show superior performance over conventional methods (Hou & Zhang, 2007; Wang et al., 2011).

Given a three-channel color image I with dimensions of $W \times H$, the trained SOD model f processes the image and outputs a binary salient map S, where $S = f(I) \in [0,1]^{W \times H}$. Specifically, in training dataset, every image I has a ground-truth map $G(I) \in [0,1]^{W \times H}$, and during the training process, the ultimate purpose is to find a model f that makes the destination between $f(I)$ and $G(I)$ to a minimum (Wang, Lai, et al., 2022). As one of the most popular and effective strategy used in SOD, multi-scale learning (MUL) method leverages low/high-level learning to better capture the fine-grained details of the object and locate the position of the salient regions, respectively (Hou et al., 2019; Liu, Han, et al., 2020). However, exiting MUL-based SOD methods still pay insufficient attention to high-level features (Liu et al., 2019; Zhao & Wu, 2019), to address this issue, EDN (Wu, Liu, Zhang, et al., 2022) used an Extremely Downsampled Block (EDB) to learn a global view of the image and achieved high accuracy with real-time speed in experiments. The architecture of the EDN is depicted in Fig. 1(a), in which scale-correlated pyramid convolution (SCPC) is constructed to effectively aggregate multi-level features from top to bottom, just as shown in Fig. 1(b). On the basis of EDN, we want to implement an ROI detection method that requires less computation resources and can quickly detect salient regions and complete sensitive regions protection. Here, the backbone of EDN is replaced by MobileNetv2 for realizing lightweight, just as depicted in Fig. 2. Furthermore, Conv3 × 3 block in EDB and Conv3 × 3 in SCPCs are replaced by inverted residual block and depth-wise separable 3 × 3 convolutions, respectively, thus making our algorithm much more practical.

2.2. Chaotic system

2.2.1. LICC hyperchaotic system

As a multi-dimensional discrete hyperchaotic system, LICC (a combination of the Logistic map and the Iterative Chaotic Map with Infinite Collapse (ICMIC) through a closed-loop coupling method) is much more complex and uncertain than other chaotic system (Wei & Li, 2022), which is defined by

$$\begin{cases} x(i+1) = \cos[c(\frac{1}{a \cdot y(i)(1-y(i))} + \frac{1}{a \cdot z(i)(1-z(i))})] \sin[\frac{b}{x(i)}], \\ y(i+1) = \cos[c(\frac{1}{a \cdot x(i)(1-x(i))} + \frac{1}{a \cdot z(i)(1-z(i))})] \sin[\frac{b}{y(i)}], \\ z(i+1) = \cos[c(\frac{1}{a \cdot x(i)(1-x(i))} + \frac{1}{a \cdot y(i)(1-y(i))})] \sin[\frac{b}{z(i)}], \end{cases} \tag{1}$$

where a, b, c are the parameters of the LICC hyperchaotic system.

The phase diagram of the chaotic system can reflect the distribution of chaotic time series. Just as shown in Fig. 3, when the number of iterations is 50,000, the chaotic sequences generated by LICC is relatively uniform in region D,

$$D = \{(x, y, z) | x \in [-1, 1], y \in [-1, 1], z \in [-1, 1]\}. \tag{2}$$

Fig. 4 is the distribution of the chaotic sequences generated by LICC hyperchaotic system when three parameters are changed respectively. Here, each parameter is traversed in steps of 0.01 from 0 to 5, and the length of the chaotic sequences is 100, thus there are a total of 50,000 scatters in each graph. It is obviously that the system can stably maintain chaos and evenly distribute in $[-1, 1]$ when the parameters change.

2.2.2. Logistic map

To achieve cross-color channel diffusion, the proposed algorithm will randomly group the subpixels of the three color channels in each pixel. After traversing all pixels, the three groups are obtained and then encrypted in parallel. Here, logistic map is introduced for grouping to modularize the pixel grouping and the subsequent encryption processes. Mathematically, logistic map is defined by

$$x_{n+1} = \lambda \cdot x_n \cdot (1 - x_n), \tag{3}$$

when $\lambda \in [3.67, 4]$, it is a chaotic system.

This modular design facilitates debugging and helps identify potential issues during the coding process. Therefore, we use logistic map, which is easy to implement and encode, to control the grouping process.

2.3. Reversible data hiding

Reversible Data hiding means insert some useful information into image, and the image can be restored to the original image after the information is extracted. In our ROI image encryption algorithm, the side information of ROI in each patch is crucial for the decryption to be successful. However, delivering this information directly over the Internet increases the risk of leakage and the burden of designing the distribution algorithm. Thus, hiding the side information into the encrypted image and sending them together is a more secure and reasonable option.

Here, a new reversible data hiding scheme proposed by Jia (Jia et al., 2019) is used to hide the side information. Compared with some existing reversible data hiding methods (Chen et al., 2013; Li et al., 2013; Thodi & Rodríguez, 2007; Tsai et al., 2009), it embeds data in the smooth regions of the image, thus reducing the distortion and improving the capacity. The detail process is described as follow.

Step 1: As shown in Fig. 5, by ignoring the border pixels, the original image is divided into two parts in a checkerboard pattern, labeled as A and B, respectively. Since the embedding operation in the two parts is the same, only the embedding data in A is described in the following steps.

Step 2: First, the local complexity Ω_p in part A is calculated as follows.

$$\Omega_p = |p_1 - p_4| + |p_2 - p_3| + |p_1 + p_3 - p_2 - p_4| + |p_3 + p_4 - p_1 - p_2|,$$

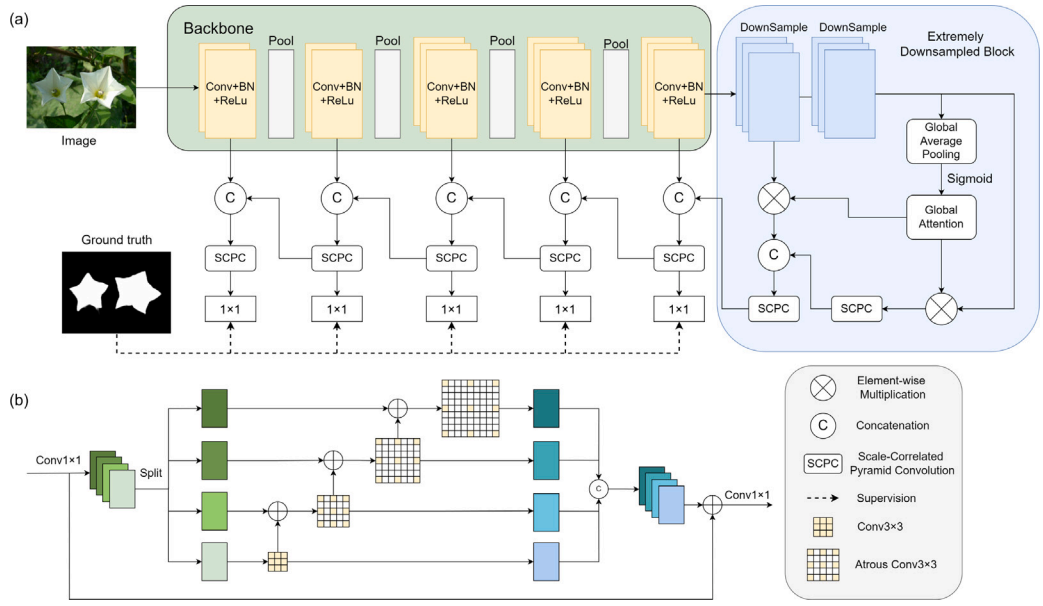


Fig. 1. The architecture of EDN.

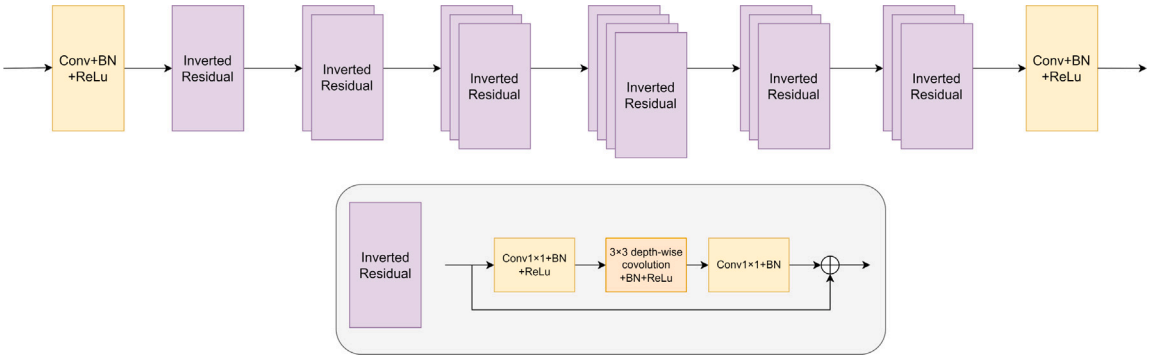
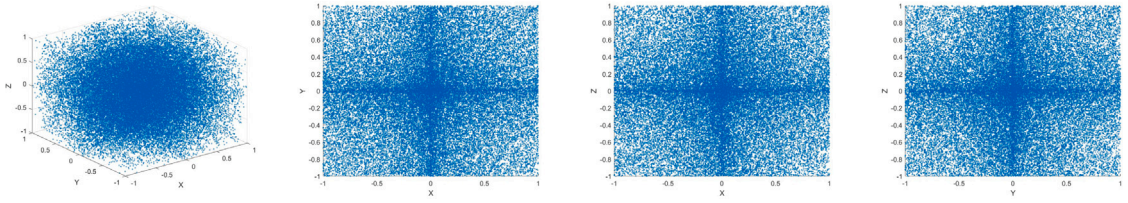
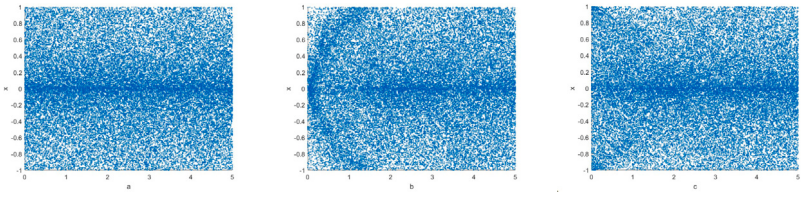


Fig. 2. The backbone of lightweight EDN.



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

Fig. 3. Attractor diagrams of LICC system with initial value $(x_0, y_0, z_0) = (0.323690874136891, 1.567982314509317, 0.314829045130345)$ and $a = 3.9, b = \pi, c = \pi$. (a) The attractor in LICC system (b). The projections of x–y planes, (c). The projections of x–z planes, (d). The projections of y–z planes.



(a) (b) (c)

Fig. 4. Bifurcation diagram of LICC system initial value $(x_0, y_0, z_0) = (0.323690874136891, 1.567982314509317, 0.314829045130345)$. (a). Bifurcation with parameter a when $b = \pi$, $c = \pi$.,(b). Bifurcation with parameter b when $a = 3.9$, $c = \pi$, (c). Bifurcation with parameter c when $a = 3.9$, $b = \pi$.

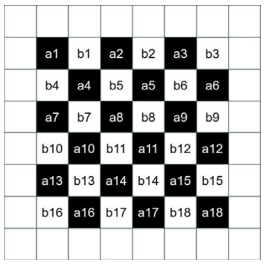


Fig. 5. Dividing the image in a checkerboard pattern, black pixels are A and white pixels are B.

where p_1, p_2, p_3, p_4 are the value of pixels directly adjacent to pixel p: above, left, right, and below, respectively.

Secondly, the calculation of the fluctuation value F_p of each pixel in part A has three situations, which are one adjacent pixel, two adjacent pixels, and four adjacent pixels. Taking $a1, a2$ and $a4$ in Fig. 5 as an example, their fluctuation values are shown below,

$$\begin{cases} F_{\Omega_{a1}} = \Omega_{a1} + \Omega_{a4}, \\ F_{\Omega_{a2}} = \Omega_{a2} + \lfloor \dfrac{\Omega_{a4} + \Omega_{a5}}{2} \rfloor, \\ F_{\Omega_{a4}} = \Omega_{a4} + \lfloor \dfrac{\Omega_{a1} + \Omega_{a2} + \Omega_{a7} + \Omega_{a8}}{4} \rfloor, \end{cases} \tag{4}$$

where $\lfloor \cdot \rfloor$ means floor function.

Step 3: The computation of prediction error e_p of each pixel is described below.

Step 3.1: First, for pixel p, its prediction value P'_p is

$$P'_p = \lfloor \omega_1 \cdot p_1 + \omega_2 \cdot p_2 + \omega_3 \cdot p_3 + \omega_4 \cdot p_4 \rfloor, \tag{5}$$

where $\omega_1, \omega_2, \omega_3, \omega_4$ are the weights of $\{p_1, p_2, p_3, p_4\}$, respectively, and $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$.

Step 3.2: The average value P_{means} of the four adjacent pixels for p is calculated as

$$P_{means} = \lfloor \frac{p_1 + p_2 + p_3 + p_4}{4} \rfloor. \tag{6}$$

Step 3.3: The difference between the values of the four adjacent pixels for p and P_{means} is calculated as follows,

$$\begin{cases} e_1 = p_1 - P_{means}, \\ e_2 = p_2 - P_{means}, \\ e_3 = p_3 - P_{means}, \\ e_4 = p_4 - P_{means}, \end{cases} \tag{7}$$

Step 3.4: According to e_1, e_2, e_3, e_4. ω'_1, ω'_2, ω'_3, ω'_4 are computed as follows,

$$\omega'_i = \begin{cases} \dfrac{1}{4}, & \sum_{j=1}^{4} |e_i| = 0 \\ \dfrac{\sum_{j=1}^{4} |e_j|}{1 + |e_i|}, & otherwise \end{cases} \tag{8}$$

where $i \in [1, 4]$, and $\omega_1, \omega_2, \omega_3, \omega_4$ are the normalization results of $\omega'_1, \omega'_2, \omega'_3, \omega'_4$ respectively.

The prediction error of pixel p is computed by

$$e_p = P_p - P'_p \tag{9}$$

where P_p is the value of pixel p.

Step 4: Continue repeating the above process, and all the extracted fluctuation value and prediction error value of each pixel are denoted as F_{seq} and P_{seq}, respectively.

Step 5: The data to be embedded is encrypted using XOR at the byte level, where the key stream is generated by another independent logistic map. The embedding process is shown in Alg. 1. Here, $data[num]$ means the num-th bit to be embedded and $I[i]$ means the ith pixel in image I.

The extraction of embedded data and the recovery of images are the reverse operations of embedding.

Algorithm 1 The embedding process

Require: $F_{seq}, P_{seq}, data, I$
Ensure: Image that contains steganography information.

```
function EmbedData(F_seq, P_seq, data, I)
    Sort F_seq and P_seq in ascending order, and the sorted sequences are denoted as F'_seq and P'_seq respectively.
    Find two peak points PK_1, PK_2 and the nearest zero point Z_1 to their left and the nearest zero point Z_2 to their right from P_seq.
    num ← 0
    for i = 0 to len(P_seq) − 1 do
        while num ≤ len(data) do
            if P'_seq[i] == min(PK_1, PK_2) then
                I[i] −= data[num].
                num++.
            else if P'_seq[i] == max(PK_1, PK_2) then
                I[i] += data[num].
                num++.
            else if Z_1 < P'_seq[i] < min(PK_1, PK_2) then
                I[i] −= 1.
            else if max(PK_1, PK_2) < P'_seq[i] < Z_2 then
                I[i] += 1.
            end if
        end while
    end for
end function
```

3. ROI-based image encryption algorithm

Fig. 6(a) depicts the encryption process of our algorithm, which comprises of the detection of sensitive region, parallel encryption and reversible data hiding. The corresponding decryption process has roughly the inverse operations, as shown in Fig. 6(b).

Here, the "semantically enhanced" aspect is applied by integrating lightweight-EDN saliency detection network as shown in Figs. 1 and 2 to identify ROI regions efficiently. This lightweight network captures essential semantic features, such as object significance and spatial relevance, while maintaining low computational complexity. By focusing on the most salient areas in an image, our method balances detection speed with accuracy. Then, with the coordinate information of the detected salient regions, the proposed method enables precise extraction of sensitive region pixels, which are then encrypted and protected. From this perspective, our ROI encryption incorporates semantic enhancement, achieving efficient ROI detection and encryption of sensitive regions.

For ease of reading, Table 2 lists the definition of parameters used in Section 3.

3.1. The detection of sensitive region

Let us take Fig. 7(a) as example, the proposed PSRD is introduced as follows.

Step 1: Input Fig. 7(a), EDN-lite outputs a binary salient map, as shown in Fig. 7(b), along with the coordinates of each sensitive region in image.

Step 2: Sensitive region analysis and detection.

Case 0: As shown in Fig. 7(c), obviously, the shape of the white regions can tell us a rough idea of its content, so directly encrypting these pixels cannot ensure the security of sensitive region.

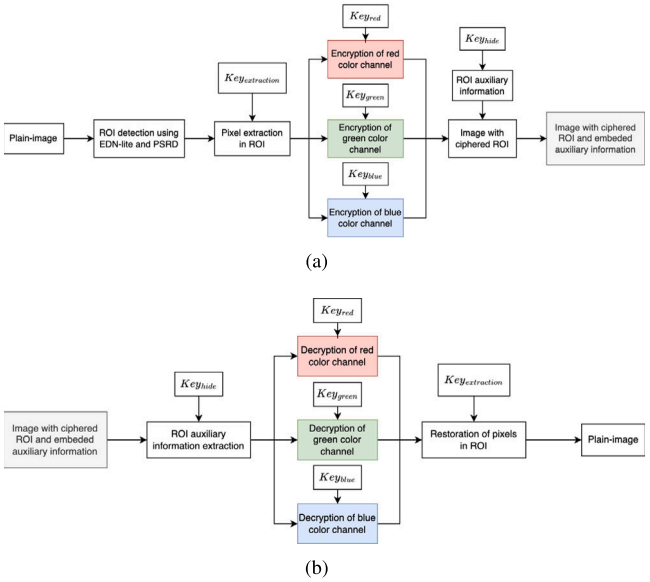


Fig. 6. (a). The flowchart of encryption scheme, (b). The flowchart of decryption scheme.

Table 2
Function parameter definition in Section 3.

Parameter	Definition
I	The plain-image
N^2	The number of patches segmented by PSRD
ROI_{info}	The side information of sensitive regions
$pixel_{num}$	The number of pixels to be encrypted
T	The number of pre-iterations of LICC and logistic map
$index'_{seq}$	The shuffled patch indexes sequence
$chaoseq_{logi}$	The chaotic sequence generated by logistic map
$chao_0, chao_1, chao_2$	The chaotic sequences generated by LICC
seq_0, seq_1, seq_2	The pixel sequences to be encrypted
$cseq_0, cseq_1, cseq_2$	The ciphered pixel sequences
$dseq_0, dseq_1, dseq_2$	The decrypted pixel sequences

Case1: To conceal shape information, a common approach is to obtain the boundary coordinates of the white region (top, bottom, left, and right) in Fig. 7(c), and then extend the white region into a rectangle based on these values as shown in Fig. 7(d). Although encrypting the region in Fig. 7(d) can meet our security requirements, the encryption of some non-sensitive regions (e.g. the regions between two horses) leads to a low encryption efficiency.

Our trade-off detection method first divides the salient map output in Step 1 evenly into $N \times N$ patches. If a patch includes white regions, it is considered to include sensitive regions. To further eliminate most of the non-sensitive regions within each patch, our method calculates the minimum value of the upper/left boundary and the maximum value of the lower/right boundary across all white regions. Then, the refined white region is shown in Fig. 7(e). Compared with Figs. 7(c) and (d), it is obvious that our method does not leak the information of the shape of the objects and has no redundant regions. Therefore, the proposed method enables efficient detection and security tagging of sensitive regions.

Step 3: The final detection output with PSRD method is shown in Fig. 7(f), and the tag information of sensitive region in each patch is stored in ROI_{info}, which contains the number of sensitive regions and the top-left and lower-right coordinates of the sensitive region.

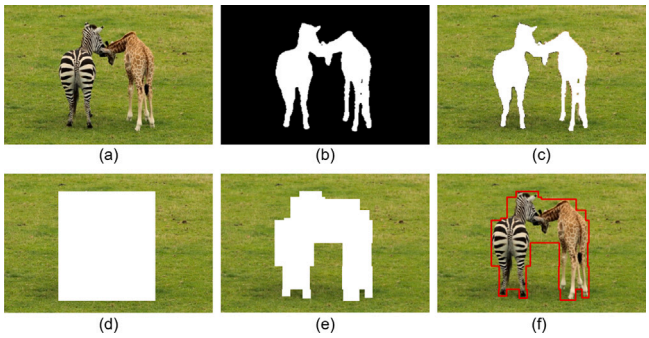


Fig. 7. (a). Original image, (b). The salient map of (a), (c). Maping (b) directly to the original image as sensitive region, (d). Covering the salient region in (b) with a rectangular box as sensitive region, (e). Leveraging the PSRD to cover the salient region as sensitive region, (f). The output of PSRD.

Table 3
Pixels allocation rule in encryption stage.

	Red	*Green*	*Blue*
$rule0$	seq_0	seq_1	seq_2
$rule1$	seq_0	seq_2	seq_1
$rule2$	seq_1	seq_0	seq_2
$rule3$	seq_1	seq_2	seq_0
$rule4$	seq_2	seq_0	seq_1
$rule5$	seq_2	seq_1	seq_0

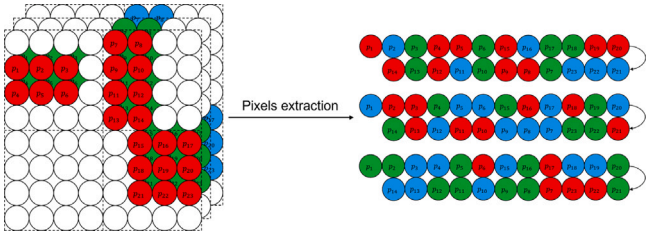


Fig. 8. The ROI pixels of an RGB image are assigned to three pixel sequences through a logistic map.

3.2. The encryption of sensitive regions

3.2.1. Pixel extraction in sensitive regions

Fig. 8 is the schematic of the extraction procedure and the corresponding detailed operations is shown in Alg. 2. It should be noted that each patch has a corresponding index, ranging from 0 to $N^2 - 1$. And $index'_{seq}$ is the sequence of shuffled patch indexes.

First, the number of pixels in sensitive regions num is calculated according to ROI_{info}. Then, logistic map is pre-iterated for T times, so that it can enter a chaotic state. Then, logistic map is iterated for $pixle_{num}/3$ times to generate chaotic sequence $chaoseq_{logi}$. During extraction, the three color components in each pixel is randomly assigned to a sequence according to the allocation rules in Table 3, and the allocation rule of each pixel is determined by $chaoseq_{logi}$, as shown in lines 4–6.

3.2.2. Multi-channel parallel encryption

Fig. 9 depicts the parallel encryption process of three extracted pixel data. It should be noted that before parallel encryption, LICC hyperchaotic system is pre-iterated T times to make it enter a chaotic state, and then iterated $len(seq_0)$ times to generate $chao_0$, $chao_1$, $chao_2$ to encrypt seq_0, seq_1, seq_2, respectively. The detailed encryption operations in each thread and thread synchronization parameter setting operations are shown in Alg. 3 and Alg. 4.

To further improve the encryption efficiency, the extracted pixels in each thread are encrypted in a simultaneous permutation–diffusion

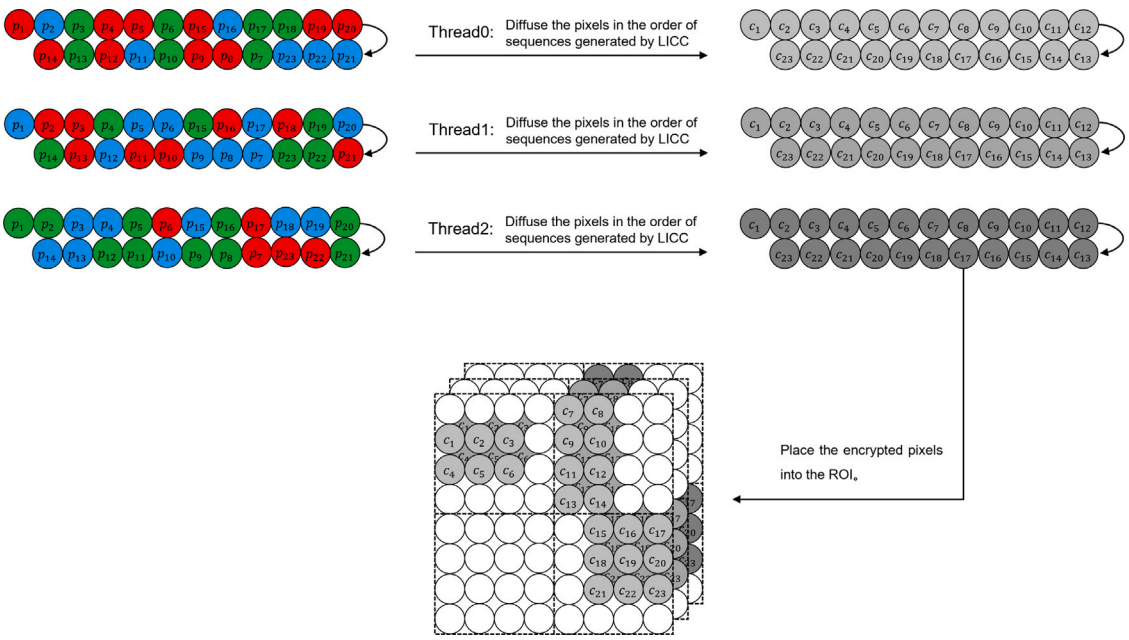


Fig. 9. The diagram of the multi-channel parallel encryption method.

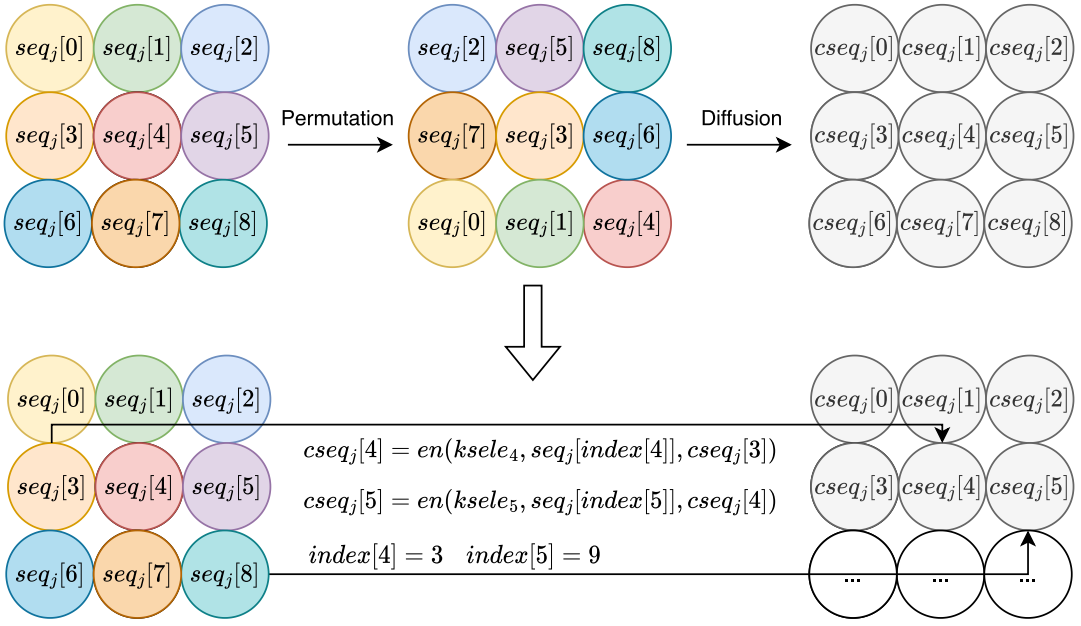


Fig. 10. The diagram of the simultaneous permutation–diffusion.

manner, as shown in Alg. 3. Here, the permutation coordinates are quantized from generated chaotic sequence as shown in lines 3–6. Specifically, the right parts in Fig. 10 depicts the simultaneous permutation–diffusion operations in lines 7–13. It can be seen that the output value of $cseq_j[i]$ is determined by the values of current keystream elements $ksele$, the current plain-pixel $seq_j[index[i]]$ and the previous ciphered pixel $cseq_j[i-1]$, respectively. Here, the introduction of permutation coordinates $index[i]$ achieves pixel permutation, while the incorporation of the previous cipher pixel simultaneously realizes the diffusion of the encryption effect. We can also visually observe from Fig. 10 that, in contrast to the upper part of Fig. 10, the strategy employed on the lower part demonstrates higher efficiency by saving one round of pixel data traversal for permutation. Additionally, L is 256 when image I is an RGB image with eight bits per channel.

Alg. 4 demonstrates operations related to thread parameter settings during parallel encryption. Specifically, the *_beginthreadex*() function from the Windows API is invoked for three times to create the three threads, and the initial state of the three threads is set to suspended. Then the priority and affinity of the three threads and thread arguments (the pixel sequences and pseudo-random chaotic sequences assigned to each thread) are set to make sure that these threads can encrypt each pixel sequence strictly on different processors. Next, the three threads are resumed to begin parallel encryption.

3.2.3. Data steganography

The related information ROI_{info} and shuffled patch index sequence $index'_{seq}$ generated in Alg. 2 are declared as integer and hidden into the encrypted image with the method introduced in Section 2.3. Here,

Algorithm 2 Pixel extraction in sensitive regions

Require: I, ROI_{info}, N
Ensure: $seq_0, seq_1, seq_2, index'_{seq}$
1: **function** PixelExtract(I, ROI_{info}, N)
2: Calculate the number of pixels $pixel_{num}$ in sensitive regions according to ROI_{info}.
3: Pre-iterate logistic map T times, and then iterate $pixel_{num}/3$ times to generate chaotic sequence, $chaoseq_{logi}$.
4: **for** $i = 0$ to $len(chaoseq_{logi}) - 1$ **do**
5: $rule[i] \leftarrow |chaoseq_{logi}[i]| \times 10^{15} \mod 6$.
6: **end for**
7: $flag \leftarrow 0, index'_{seq} \leftarrow [\cdot]$
8: **for** $i = 0$ to $N \times N - 1$ **do**
9: **if** $flag \in index'_{seq}$ **then**
10: $index_{seq} \leftarrow [0, 1, ...flag]$
11: $flag$ is changed to the maximum value of set $index_{seq} - index'_{seq}$ that is less than the $flag$, or the minimum value that is greater than the $flag$.
12: **end if**
13: **if** $ROI_{info}[flag].num \neq 0$ **then**
14: All the ROI pixels of $patch[flag]$ were extracted and randomly assigned to three pixel sequences according to $rule$.
15: **end if**
16: $index'_{seq}.append(flag)$
17: $patch_{temp} \leftarrow patch[i]$
18: $flag \leftarrow patch_{temp}[-1][-1][-1] \mod (N \times N)$
19: **end for**
20: **end function**

Algorithm 3 The encryption operations in $thread_j$

Require: $chao_j, seq_j, j \in \{0, 1, 2\}$
Ensure: $cseq_j, j \in \{0, 1, 2\}$
1: **function** Encryption($chao_j, seq_j, j \in \{0, 1, 2\}$)
2: $index \leftarrow [0, 1, 2..., len(seq_1) - 1]$
3: **for** $i = 0$ to $len(seq_1) - 1$ **do**
4: $chao_{Val} \leftarrow (int64)(|chao_j[i]| \cdot 10^{15})\%(len(seq_j) - i)$
5: Swap the value of $index[chao_{Val}]$ and $index[len(seq_j) - 1 - i]$.
6: **end for**
7: $preVal \leftarrow (int64)(|chao_j[1]| \cdot 10^{15})\%L$
8: $ksele \leftarrow (int64)(|chao_j[0]| \cdot 10^{15})\%L$
9: $cseq_j[0] \leftarrow ksele \oplus [(seq_j[index[0]] + ksele)\%L] \oplus preVal$
10: **for** $i = 1$ to $len(seq_j) - 1$ **do**
11: $ksele \leftarrow (int64)(|chao_j[i]| \cdot 10^{15})\%L$
12: $cseq_j[i] \leftarrow ksele \oplus [(seq_j[index[i]] + ksele)\%L] \oplus cseq_j[i - 1]$
13: **end for**
14: **end function**

Algorithm 4 The parallel encryption of extracted pixel data

Require: $seq_j, chao_j, ROI_{info}, j \in \{0, 1, 2\}$
Ensure: The ciphered pixel data.
1: **function** ParallelEncryption($seq_j, chao_j, ROI_{info}, j \in \{0, 1, 2\}$)
2: Create three threads and suspend them.
3: Set the affinity and arguments of each thread.
4: Resume the execution of threads.
5: **spawn**
6: Encryption($chao_j, seq_j$)
7: **sync**
8: According to ROI_{info}, write the ciphered pixel data back to image.
9: **end function**

Table 4
Pixels allocation rule in decryption stage.

	seq_0	seq_1	seq_2
$rule0$	*Red*	*Green*	*Blue*
$rule1$	*Red*	*Blue*	*Green*
$rule2$	*Green*	*Red*	*Blue*
$rule3$	*Blue*	*Red*	*Green*
$rule4$	*Green*	*Blue*	*Red*
$rule5$	*Blue*	*Green*	*Red*

each patch uses 1 bit to indicate sensitive regions: 0 for none, 1 for present. The coordinate information of the sensitive region in each patch need 32 bits to represent, and the index of each patch needs 6 bits to represent.

When embedding data, we first store the above data into an array in units of bits. Then, for the image I in which data is to be embedded, we perform **Steps** 1 to 4 from Section 2.3 to obtain F_{seq} and P_{seq}. Finally, we perform **Step** 5 and invoke Alg.1 to embed the data into image I.

By using steganography, we add an additional layer of protection to the ROI side information and complete the transmission of all data with just a single data distribution.

3.3. Decryption process

The decryption process is roughly the reverse of the encryption process, which contains the extraction of encrypted ROI pixels, the decryption of three ROI pixel sequences in parallel, and the use of the rule in Table 4 to restore these pixels to their original channels. Particularly, the reverse of line 12 in Alg. 3 is given by

$$dseq_j[index[i]] = [(seq[i] \oplus seq[i-1] \oplus ksele) + L - ksele]\%L, j \in \{0, 1, 2\} \tag{10}$$

where for an RGB images with eight bits per channel, $L = 256$.

4. Experimental results and security analysis

In experiments, different images are encrypted to test the performance of the proposed ROI encryption method, in which the initial value and parameters of LICC are set to $(0.3, 1.5, 0.9)$ and $(3.9, \pi, \pi)$ respectively, and N in PSRD method is set to 6 unless otherwise specified.

4.1. Key space analysis

In the proposed scheme, the encryption keys, which correspond to the initial values of the LICC system, are randomly assigned by users within the predefined range of the initial values of LICC system. This allows for flexibility while ensuring that the assigned keys meet the necessary conditions for secure encryption. Therefore, the key space contains three double-precision floating-point numbers and each value has 15 bits effective accuracy. As a result, the key space is $(10^{15})^3$, which is approximately equal to 2^{150} and larger than 2^{100}. Therefore, the method is secure enough to resist brute force attack (Alvarez & Li, 2006).

4.2. Statistical properties analysis

Due to the inherent visual redundancy of images, statistical attacks have evolved into the most common and efficient method for attacking image encryption systems. In this part, histogram analysis, correlation of adjacent pixels analysis and information entropy analysis are used to prove the security of the algorithm when facing statistical attack.

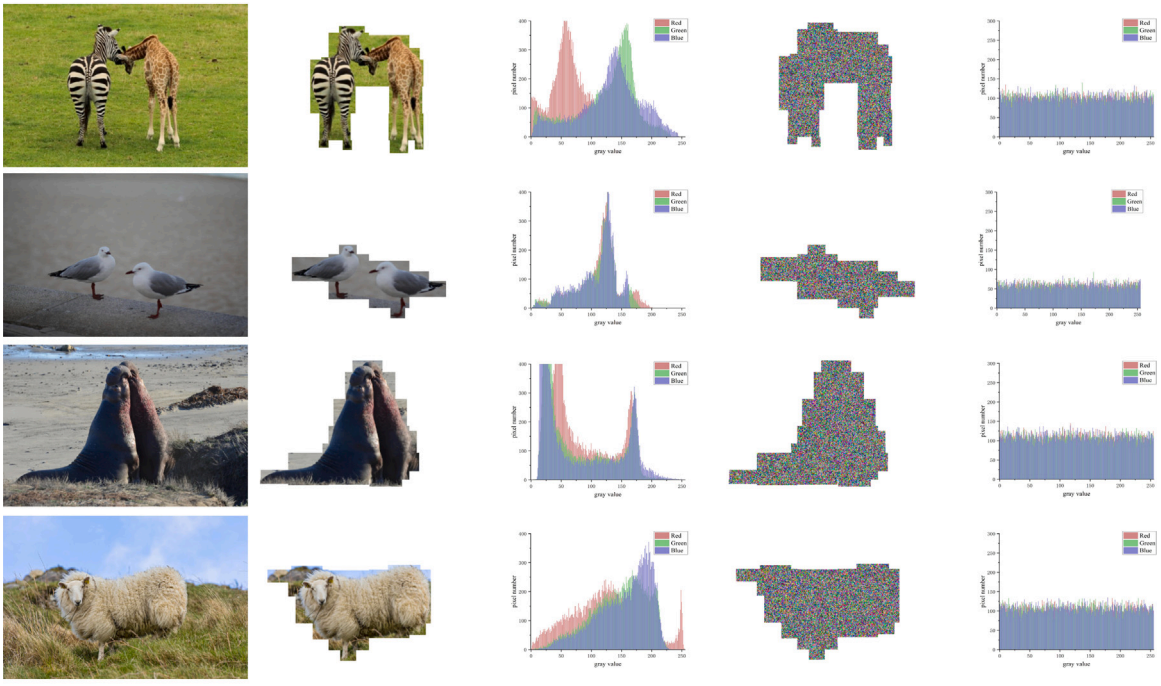


Fig. 11. (a). Original images, (b). ROI before encryption, (c) Histograms of ROI before encryption, (d). ROI after encryption, (e). Histograms of ROI after encryption.

Table 5
Information entropy of four plain images and encrypted images.

Image	Plain region	Encrypted region
Zebra & giraffe	7.4347	7.9933
Bird	6.9003	7.9870
Walrus	7.0042	7.9935
Sheep	7.4948	7.9934

4.2.1. Histogram analysis

In order to defend statistical attack, the histogram of the encrypted regions should be as uniform as possible. Fig. 11 shows histograms of the ROI of four images before and after encryption, here, the histograms after encryption are evenly distributed, which demonstrates that the pixel distribution in sensitive region is visually hidden.

4.2.2. Information entropy analysis

Information entropy is used to measure the randomness of an image, which is defined by

$$H(x) = -\sum_{i=1}^{n} P(x_i) log_2(P(x_i)), \tag{11}$$

where, n is the number of pixels and $P(x_i)$ is the value of pixel x_i. We use images in Fig. 11(a) as test images, whose gray level is 256, so the information entropy of encrypted regions should be close to 8. From Table 5, the information entropy of four encrypted regions are all close to 8, this result is directly linked to the chaotic map employed in our encryption process. The inherent randomness of the LICC and logistic map results in highly unpredictable encryption patterns. Furthermore, our algorithm strategically integrates chaos theory into both the encryption framework, enhancing the system's ability to generate secure and random patterns. As demonstrated by the high entropy values in Table 5, these design choices ensure that the cipher-regions achieves near-maximal randomness and robustness.

4.2.3. Correlation of adjacent pixels analysis

In plain-images, adjacent pixels often exhibit strong correlation, which can be exploited by attackers to infer pixel values. An effective encryption algorithm must have the ability to break this correlation to enhance the randomness and security of the encrypted image, making it more resistant to cryptanalysis (Song, Fu, Lin, et al., 2024). To analysis the correlation of adjacent pixels in encrypted region, just as shown in Fig. 12, we set the N in PSRD method to 1 to get a rectangular ROI and select 1000 pairs of pixels from it randomly.

Fig. 12. (a). Original image, (b). Encrypted image when N is 1.

Fig. 13 shows the visualization of the correlation of horizontal, vertical and diagonal adjacent pixels in ROI in Figs. 12(a) and 12(b), where the red, green and blue dots represent the correlation of the image on the R, G and B components, respectively. For the original image, most points are concentrated near the diagonal of each y–z plane, indicating a strong correlation between adjacent pixels. For the encrypted image, points are distributed in the whole y–z plane evenly, showing that the high correlation of the original image has been broken (Chen et al., 2020; Chen, Guo, et al., 2023; Chen, Wang, et al., 2023; Gong & Luo, 2023).

The correlation coefficients can quantitatively assess the relevance of adjacent pixels, which is shown in Eq. (12).

$$r(x,y) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{(\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2)(\frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2)}}, \tag{12}$$

where x_i, y_i are the values of adjacent pixels, N is the number of pixels, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$, and $E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i$.

Table 6 shows the correlation coefficients between the R, G, and B components of the plain-region Fig. 12(a) and the corresponding

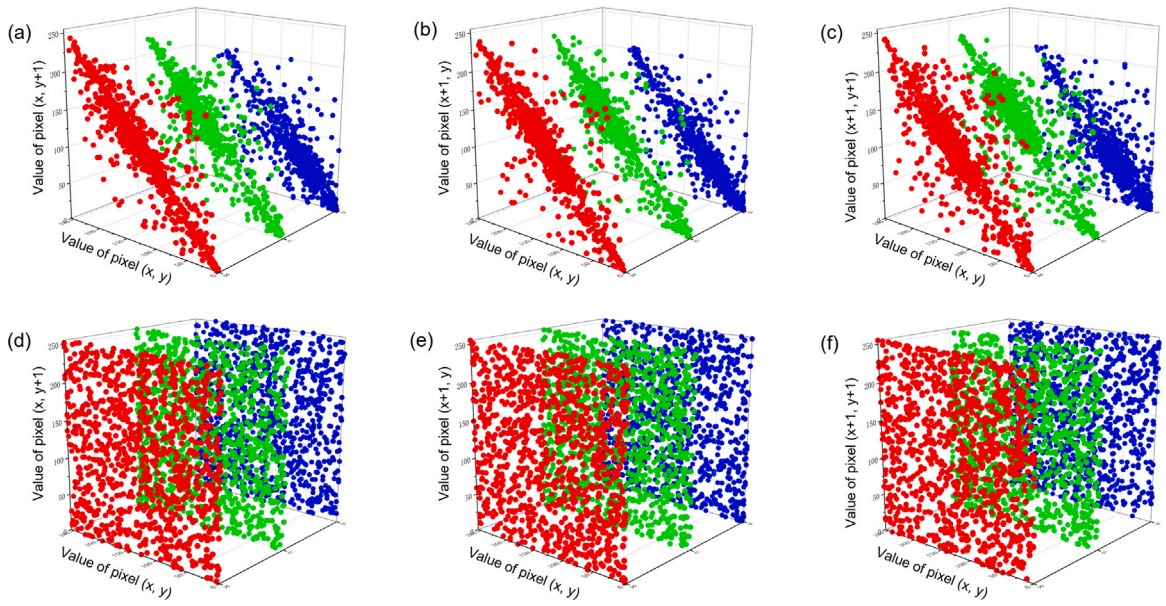


Fig. 13. (a), (b), (c) are the scatter diagrams of the horizontal, vertical and diagonal adjacent pixels in ROI of the original image respectively. (d), (e), (f) are the scatter diagrams of the horizontal, vertical and diagonal adjacent pixels in ROI of encrypted image respectively.

Table 6
Correlation coefficients of the plain-image and the corresponding cipher-image.

Direction	Plain region			Encrypted region		
	R	*G*	*B*	*R*	*G*	*B*
Horizontal	0.822613	0.848189	0.817465	0.013102	0.035463	0.016454
Vertical	0.851788	0.872268	0.821505	0.004168	0.004498	−0.018506
Diagonal	0.735205	0.772554	0.719809	0.043012	0.012318	0.015241

cipher-region Fig. 12(b) in the horizontal, vertical, and diagonal directions. From this table, we find that the correlation in encrypted image is close to 0, which implies our method can decorrelate the strong correlation in each cipher region. The low correlation between adjacent pixels in these cipher-regions stems from the integration of chaos theory and our cross-color channel encryption design. These features disrupt spatial relationships, ensuring the cipher-regions effectively obscure patterns and enhances security.

4.3. Differential attack

In order to resist differential attack, two images with only minor differences should be completely different after encryption. $NPCR$ (the number of pixel change rate) and $UACI$ (the unified average changing intensity) are two criteria for measuring the difference between two regions with the same size. For the image whose gray level is 256, the theoretical values of $NPCR$ and $UACI$ are 99.609% and 33.464%, respectively (Guo et al., 2024). As shown in Fig. 11, our detection results always have irregular regions, but measuring the difference between them is complex. As previously described, during the encryption process, we extract the pixels from the irregular ROI regions into a one-dimensional array. Therefore, we perform an equivalent transformation of $NPCR$ and $UACI$ to measure the difference between two one-dimensional arrays. The transformed formulas are as follows:

$$NPCR = \frac{\sum_{i=1}^{L}(D(i))}{L} \times 100\%, \tag{13}$$

$$UACI = \frac{1}{L}\left(\sum_{i=1}^{L}\frac{|P_1(i) - P_2(i)|}{255}\right) \times 100\%, \tag{14}$$

Table 7
Result of $NPCR$ and $UACI$ tests.

$NPCR$	$UACI$
99.5759%	33.3449%

Table 8
Different secret keys.

	x_0	y_0	z_0
key_1	0.323690874136891	1.567982314509317	0.314829045130345
key_2	0.323690874136892	1.567982314509317	0.314829045130345
key_3	0.323690874136891	1.567982314509318	0.314829045130345
key_4	0.323690874136891	1.567982314509317	0.314829045130346

where L is the length of the array, $P_1(i)$ and $P_2(i)$ are the values of the two arrays at position i, and

$$D(i) = \begin{cases} 0 & P_1(i) = P_2(i), \\ 1 & P_1(i) \neq P_2(i). \end{cases} \tag{15}$$

We use secret key to encrypt Fig. 12(a), then, the pixel value of (37, 21) in Fig. 12(a) is added by one and the new image is encrypted with the same key, last, calculate the $NPCR$ and $UACI$ of ROI in two encrypted images. The test result is shown in Table 7, which is close to the theoretical values.

4.4. Key sensitive analysis

An excellent encryption algorithm should be sensitive to secret key, which means a minor change in the key should produce a completely different result.

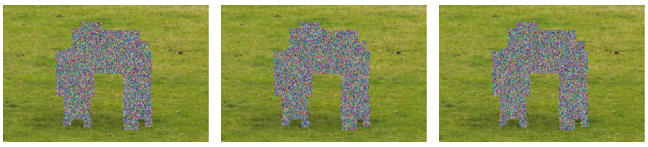
4.4.1. Encryption key sensitive analysis

Four keys listed in Table 8 are used to encrypt Fig. 12(a), here, different keys only have minor different in the least significant bit.

We use key_2, key_3 and key_4 as modified keys, and calculate $NPCR$ and $UACI$ between the encrypted region encrypted by modified keys and the encrypted region encrypted by key_1. Just as shown in Table 9, the regions produced by different keys are completely different.

Table 9
Key sensitive analysis of encryption process.

	$NPCR$	$UACI$
$key_1\&key_2$	99.6116%	33.3105%
$key_1\&key_3$	99.6409%	33.5311%
$key_1\&key_4$	99.5835	33.3671%



(a) (b) (c)

Fig. 14. (a), (b), (c) are the decrypted results using key_2, key_3 and key_4 respectively.

Table 10
Comparison of computational efficiency.

Network	Params(M)	FLOPs(G)
YOLOv3 (Redmon & Farhadi, 2018)	61.5	193.9
YOLOv4 (Bochkovskiy et al., 2020)	52.5	119.8
U2Net (Qin et al., 2020)	44.0	235.3
EDN (Wu, Liu, Zhang, et al., 2022)	42.8	56.7
MobileSal (Wu, Liu, Xu, et al., 2022)	3.5	4.1
EDN-lite	1.8	3.1

4.4.2. Decryption key sensitive analysis

In decryption process, key_2, key_3 and key_4 listed in Table 8 are used to decrypt the image encrypted by key_1. Just as shown in Fig. 14, all keys cannot decrypt correctly, which means our algorithm is sensitive to secret key.

4.5. Comparison of performance with the SOTA methods

4.5.1. Comparison of ROI detection efficiency

As shown in Table 10, the detection efficiency of EDN-lite is compared with YOLOv3 used in Asgari-Chenaghlu et al. (2021), Sheela and Suresh (2024), Singh, Singh, et al. (2022) and YOLOv4 used in Song et al. (2022) and some SOTA deep SOD methods, such as U2Net (Qin et al., 2020), EDN (Wu, Liu, Zhang, et al., 2022) and even lightweight RGB-D SOD MobileSal (Wu, Liu, Xu, et al., 2022). It can be seen that EDN-lite EDN-lite has fewer model parameters and achieves reduced FLOPs.

Furthermore, the detected ROI should be as small as possible while ensuring the security to improve encryption efficiency. Table 11 reports the percentage of detected ROI in the entire image as shown in Fig. 15. We can see that the ROI generated by proposed PSRD is much less than other methods, demonstrating that our detection method can reduce the data to be encrypted and improve the encryption efficiency in the subsequent encryption stage.

The above analyses shows that our proposed method effectively optimizes the regions detected by EDN-lite with different values of N. This unique combination of fine-grained sensitive regions and preserved efficiency makes EDN-lite, when integrated with our PRSD approach, an optimal solution for encryption-oriented applications.

4.5.2. Comparison of ROI detection accuracy

In our ROI detection method, the accuracy is determined by EDN-lite and its performance is compared with some classic SOD models. Here, the dataset is DUTS (Wang et al., 2017), and the training is conducted using Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 10^{-4} and batch size is 24. Table 12 illustrates the test results of EDN-lite and some other lightweight networks on DUTS-TE. Here, F_β is the weighted harmonic mean of precision and recall and

Table 11
The proportion of the ROI to the entire image.

Methods	The percentage of ROI
Singh, Singh, et al. (2022)	
Sheela and Suresh (2024)	55.18%
Asgari-Chenaghlu et al. (2021)	
Song et al. (2022)	30.88%
PSRD(N=2)	30.50%
PSRD(N=4)	25.46%
PSRD(N=6)	25.36%

Table 12
Comparison of ROI detection accuracy.

Method	$F_\beta \uparrow$	$MAE \downarrow$
MobileNetV2 (Sandler et al., 2018)	0.798	0.070
ShuffleNetV2 (Ma et al., 2018)	0.789	0.071
SAMNet (Liu et al., 2021)	0.835	0.058
EDN-lite	0.862	0.045

Table 13
Comparison of encryption time(ms) between parallel and serial processing.

Image size	Parallel manner	Serial manner	Acceleration percentage
$256 \times 256 \times 3$	6.74	8.22	18.00%
$512 \times 512 \times 3$	18.40	33.44	44.97%
$1024 \times 1024 \times 3$	76.48	167.09	54.22%

MAE measures the difference between the predicted result P and the ground truth G, which are defined as,

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \tag{16}$$

$$\text{MAE}(P, G) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} |P_{ij} - G_{ij}|. \tag{17}$$

where β^2 is set to 0.3. It can be seen that EDN-lite has higher accuracy.

4.5.3. Encryption efficiency analysis

Table 13 reports the encryption time for three rounds of encryption between parallel and serial processing for images of varying sizes. The comparison results demonstrate that the parallel approach achieves a shorter runtime compared to the serial approach. And we can observe that as the data size increases, the acceleration effect becomes more pronounced. Therefore, the proposed algorithm is particularly suitable for encrypting sensitive regions in high-resolution images.

Table 14 shows the comparison results of encryption efficiency when image size is $3MN$, here, iteration of chaotic system, its time series quantization, and permutation and diffusion operations are three main time-consuming parts. In our work, permutation and diffusion are operated at the same time, and three threads encrypt pixel sequence in parallel, thus the time complexity is $\Theta(MN)$, which is faster than the other three state-of-the-art algorithms. It is noted that in Wang et al. (2023), diffusion operation contains global diffusion and crossover random diffusion, therefore, the time complexity is $\Theta(6MN)$. From the table, it is obvious that the proposed algorithm provides the best efficiency.

4.5.4. Comparison of security performance with the SOTA methods

Table 15 lists the performance comparison between our work and other SOTA (state-of-the-art) encryption algorithms. Since the number of pixels encrypted in ROI is relatively small, there is a slight gap between our test results and the SOTA methods. To begin with, the previous performance tests have demonstrated that our algorithm can theoretically fulfill the security requirements. In addition, Tables 11 and 14 indicate the efficiency advantage of our work.

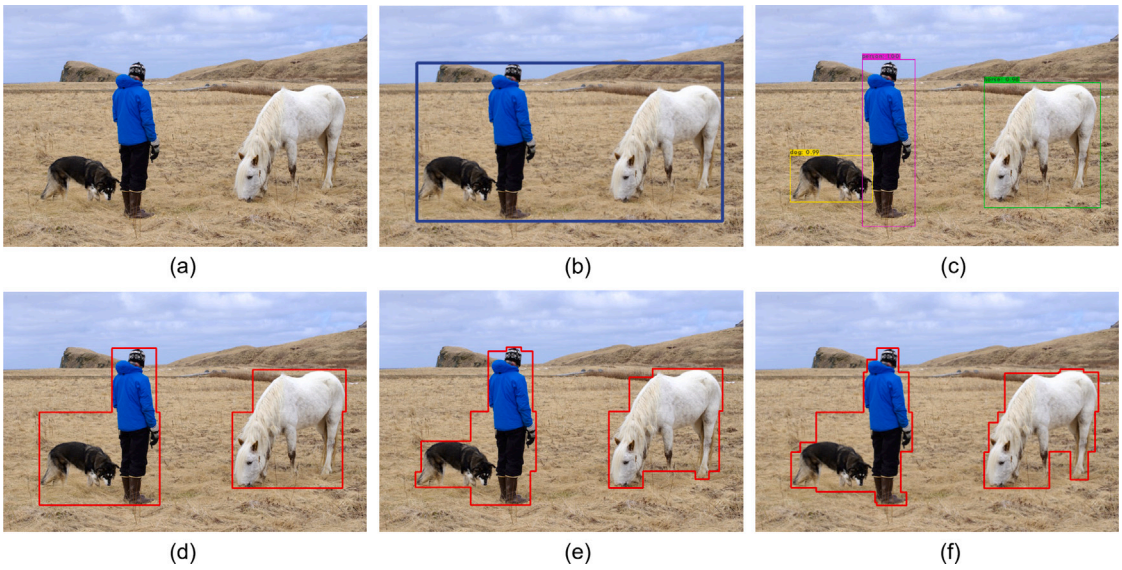


Fig. 15. (a). Original image. (b). ROI generated by YOLOv3 in Singh, Singh, et al. (2022). (c). ROI generated by modified YOLOv4 in Song et al. (2022). (d). ROI generated by PSRD(N = 2). (e). ROI generated by PSRD(N = 4). (f). ROI generated by PSRD(N = 6).

Table 14
Comparison of time complexity.

Methods	Keystream generation		Permutation operation	Diffusion operation
	Permutation stage	Diffusion stage		
Our work	$\Theta(3MN)$		$\Theta(MN)$	
Wang et al. (2023)	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(6MN)$
Singh, Singh, et al. (2022)	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$
Asgari-Chenaghlu et al. (2021)	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$
Song et al. (2022)	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$	$\Theta(3MN)$

Table 15
Comparison of security performance with the SOTA methods.

Methods	Information entropy of cipher-regions	Correlation coefficients of cipher-regions			NPCR	UACI
		Horizontal	Vertical	Diagonal		
Our work	7.9918	0.014518	−0.009898	0.011530	99.600%	33.469%
Wang et al. (2023)	7.9993	−0.000215	−0.000045	0.000037	99.609%	33.464%
Singh, Singh, et al. (2022)	7.9881	0.008400	0.019967	−0.012270	99.626%	33.449%
Asgari-Chenaghlu et al. (2021)	7.9977	−0.002130	0.001350	0.003140	99.717%	33.519%
Song et al. (2022)	7.9906	−0.001132	0.002418	−0.000632	99.392%	33.380%

Table 16
The embedding capacity of each image and the size of ROI side information under different values of N in our method, both measured in bits.

	Capacity	$N = 2$	$N = 4$	$N = 6$
Zebra & giraffe	17 690	156	368	924
Bird	64 279	124	336	572
Walrus	21 743	156	400	860
Sheep	36 914	156	432	796

Table 17
The test results of visual hiding quality.

	$PSNR$	$MSSIM$
Zebra & giraffe	69.86 dB	0.9999
Bird	79.24 dB	0.9999
Walrus	77.04 dB	0.9999
Sheep	79.48 dB	0.9999

4.6. Image steganography and ROI side information protection analysis

4.6.1. Image steganography analysis

Here, we conduct embedding capacity experiments to verify whether there is sufficient capacity to embed the ROI side information. Table 16 presents the embedding capacity of each image and the size of ROI side information under different values of N, both measured in bits. It can be observed that the spatial capacity of each image is sufficient enough to accommodate the embedding of ROI side information within our proposed method.

An effective steganographic algorithm should ensure that the image remains virtually unchanged before and after data embedding. The PSNR (Peak Signal-to-Noise Ratio) and MSSIM (Mean Structural Similarity Index) comparison results of images before and after embedding are presented in Table 17. Mathematically, the two metrics are defined as follows.

$$\begin{cases} MSE = \dfrac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P(i,j) - I(i,j))^2, \\ PSNR = 10 \log_{10}(\dfrac{255 \times 255}{MSE}). \end{cases} \tag{18}$$

$$MSSIM = \frac{1}{m} \sum_{k=0}^{m} \frac{2(\mu_P \mu_I + C_1)(2\sigma_{PI} + C_2)}{(\mu_P^2 + \mu_I^2 + C_1)(\sigma_P^2 + \sigma_I^2 + C_2)} (P_k, I_k), \tag{19}$$

where, μ_P, μ_I, σ_P and σ_I are the mean value and the standard deviation of two images, respectively. C_1 and C_2 are used to prevent the denominator from becoming zero, which are set to 6.5 and 58.5. From Table 17, it can be seen that PSNR is greater than 40 dB and MSSIM is close to 1, it can be stated that the two images are basically identical (Ye & Guo, 2024). The main reason is that the amount of data to be embedded is relatively small and is embedded only in the least significant bit of each pixel, resulting in minimal impact on visual quality.

4.6.2. ROI side information protection analysis

Table 18 illustrates the data distribution cost under scenarios with/without ROI side information hiding. It is evident that distributing extra ROI side information adds to the data transmission burden. Our steganographic mechanism not only avoids this issue but also adds an additional layer of protection to the encryption scheme. Moreover, as the volume of images increases, the savings in distribution costs become even more substantial.

4.7. Limitations

Our scheme is designed for the encryption of ROI within a single image. When encrypting the ROI of multiple images, there are certain limitations in achieving diffusion of the encryption effect across different ROI. And the number of iterations of a chaotic system in the proposed scheme is equal to the number of pixels to be encrypted. Therefore, the iteration process is time consuming when encrypting multiple images.

5. Conclusion and future work

In this paper, we propose a semantic-based image encryption algorithm to protect the ROI of the image. First, our detection method considers the security of image information and the area of the ROI comprehensively, which minimizes the encryption area as much as possible while ensuring that the image information is not leaked. Then, the pixels are allocated to three different pixel sequences randomly and encrypted in parallel to improve efficiency. After that, ROI side information is embedded into the whole image using reversible data hiding. The experimental results and security analyses show that our proposed encryption scheme is secure and convenient.

The choice of the value for N in the proposed PSRD requires further exploration, as there may be an optimization problem involved. When N is set to the optimal value, the best-optimized ROI region can be achieved. Compared to CNN-based object detection networks, studies (Han et al., 2022; Khan et al., 2022) have shown that ViT(Vision Transformer)-based models achieve higher accuracy. However, ViTs are relatively inefficient (Zheng et al., 2023). Developing an ROI encryption algorithm based on lightweight ViT for object detection is an important direction for future research.

CRediT authorship contribution statement

Buyu Liu: Conceptualization, Methodology, Software, Writing – review & editing. **Wei Song:** Conceptualization, Methodology, Software, Writing – review & editing. **Mingyi Zheng:** Conceptualization, Writing – review & editing. **Chong Fu:** Conceptualization, Methodology, Writing – review & editing. **Junxin Chen:** Methodology, Writing – review & editing. **Xingwei Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Nos. 62171114 and 62032013), the Fundamental Research Funds for the Central Universities, China (Nos. N2424010-18 and N2316010), Liaoning Provincial Science and Technology Plan Project, China (2023JH2/101700370), and the Joint Funds of Natural Science Foundation of Liaoning Province, China (No. 2023-BSBA-121).

Data availability

Data will be made available on request.

Table 18
Comparison of data distribution costs (KB) with and without ROI side information hiding.

Imgae	With hiding		Without hiding	
	Cipher-image	ROI side information	Cipher-image	ROI side information
Zebra & giraffe	306.28	0	306.28	1.13
Bird	306.28	0	306.28	0.78
Walrus	306.28	0	306.28	1.06
Sheep	306.28	0	306.28	1.00

References

Alvarez, G., & Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *International Journal of Bifurcation and Chaos*, *16*(08), 2129–2151.

Asgari-Chenaghlu, M., Feizi-Derakhshi, M.-R., Nikzad-Khasmakhi, N., Feizi-Derakhshi, A.-R., Ramezani, M., Jahanbakhsh-Nagadeh, Z., Rahkar-Farshi, T., Zafarani-Moattar, E., Ranjbar-Khadivi, M., & Balafar, M.-A. (2021). Cy: Chaotic yolo for user intended image encryption and sharing in social media. *Information Sciences*, *542*, 212–227.

Balasamy, K., & Suganyadevi, S. (2021). A fuzzy based ROI selection for encryption and watermarking in medical image using DWT and SVD. *Multimedia Tools and Applications*, *80*, 7167–7186.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. ArXiv, arXiv:2004.10934.

Cai, C., Wang, Y., Cao, Y., Sun, B., & Mou, J. (2024). Multiple remote sensing image encryption scheme based on saliency extraction and magic cube circular motion. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 5944–5960.

Chen, J., Chen, L., & Zhou, Y. (2020). Cryptanalysis of a DNA-based image encryption scheme. *Information Sciences*, *520*, 130–141.

Chen, J., Guo, Z., Xu, X., Zhang, L.-b., Teng, Y., Chen, Y., Woźniak, M., & Wang, W. (2023). A robust deep learning framework based on spectrograms for heart sound classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Chen, X., Sun, X., Sun, H., Zhou, Z., & Zhang, J. (2013). Reversible watermarking method based on asymmetric-histogram shifting of prediction errors. *Journal of Systems and Software*, *86*(10), 2620–2626.

Chen, J., Wang, W., Fang, B., Liu, Y., Yu, K., Leung, V. C. M., & Hu, X. (2023). Digital twin empowered wireless healthcare monitoring for smart home. *IEEE Journal on Selected Areas in Communications*, *41*(11), 3662–3676.

Chen, T.-H., & Yang, C.-H. (2022). Region of interest encryption based on novel 2D hyperchaotic signal and bagua coding algorithm. *IEEE Access*, *10*, 82751–82765.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, arXiv:2010.11929.

Gong, L.-H., & Luo, H.-X. (2023). Dual color images watermarking scheme with geometric correction based on quaternion FrOOFMMs and LS-SVR. *Optics and Laser Technology*, *167*, Article 109665.

Guo, Z., Chen, S.-H., Zhou, L., & Gong, L.-H. (2024). Optical image encryption and authentication scheme with computational ghost imaging. *Applied Mathematical Modelling*, *131*, 49–66.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. (2022). A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 87–110.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hou, Q., Cheng, M.-M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. S. (2019). Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(4), 815–828.

Hou, X., & Zhang, L. (2007). Saliency detection: A spectral residual approach. In *2007 IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Hu, L.-L., Chen, M.-X., Wang, M.-M., & Zhou, N.-R. (2024). A multi-image encryption scheme based on block compressive sensing and nonlinear bifurcation diffusion. *Chaos, Solitons & Fractals*, *188*, Article 115521.

Hua, Z., Zhang, Y., & Zhou, Y. (2020). Two-dimensional modular chaotification system for improving chaos complexity. *IEEE Transactions on Signal Processing*, *68*, 1937–1949.

Hua, Z., Zhu, Z., Chen, Y., & Li, Y. (2021). Color image encryption using orthogonal latin squares and a new 2D chaotic system. *Nonlinear Dynamics*, *104*, 4505–4522.

Jia, Y., Yin, Z., Zhang, X., & Luo, Y. (2019). Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting. *Signal Processing*, *163*, 238–246.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys*, *54*(10s), 1–41.

Kiran, P., & Parameshachari, B. (2022). Resource optimized selective image encryption of medical images using multiple chaotic systems. *Microprocessors and Microsystems*, *91*, Article 104546.

Kiran, Parameshachari, B., Panduranga, H., & liberata Ullo, S. (2020). Analysis and computation of encryption technique to enhance security of medical images. *IOP Conference Series: Materials Science and Engineering*, *925*(1), Article 012028.

Kocak, O., Erkan, U., Toktas, A., & Gao, S. (2024). PSO-based image encryption scheme using modular integrated logistic exponential map. *Expert Systems with Applications*, *237*, Article 121452.

Lai, Q., Lai, C., Zhang, H., & Li, C. (2022). Hidden coexisting hyperchaos of new memristive neuron model and its application in image encryption. *Chaos, Solitons & Fractals*, *158*, Article 112017.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, X., Li, J., Li, B., & Yang, B. (2013). High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion. *Signal Processing*, *93*(1), 198–205.

Liu, N., Han, J., & Yang, M.-H. (2020). PiCANet: Pixel-wise contextual attention learning for accurate saliency detection. *IEEE Transactions on Image Processing*, *29*, 6438–6451.

Liu, J.-J., Hou, Q., Cheng, M.-M., Feng, J., & Jiang, J. (2019). A simple pooling-based design for real-time salient object detection. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 3912–3921).

Liu, Y., Zhang, X.-Y., Bian, J.-W., Zhang, L., & Cheng, M.-M. (2021). SAMNet: Stereoscopically attentive multi-scale network for lightweight salient object detection. *IEEE Transactions on Image Processing*, *30*, 3804–3814.

Liu, Y., Zhang, J., Han, D., Wu, P., Sun, Y., & Moon, Y. S. (2020). A multidimensional chaotic image encryption algorithm based on the region of interest. *Multimedia Tools and Applications*, *79*, 17669–17705.

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Computer vision – ECCV 2018* (pp. 122–138). Springer International Publishing.

Murali, P., Niranjana, G., Paul, A. J., & Muthu, J. S. (2023). Domain-flexible selective image encryption based on genetic operations and chaotic maps. *Visual Computer*, *39*(3), 1057–1079.

Pang, Y., Zhao, X., Zhang, L., & Lu, H. (2020). Multi-scale interactive network for salient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9410–9419.

Peng, Y., Fu, C., Cao, G., Song, W., Chen, J., & Sham, C.-W. (2024). JPEG-compatible joint image compression and encryption algorithm with file size preservation. *ACM Transactions on Multimedia Computing, Communications and Applications*, *20*(4), 1–20.

Priyanka, Baranwal, N., Singh, K. N., & Singh, A. K. (2024). YOLO-based ROI selection for joint encryption and compression of medical images with reconstruction through super-resolution network. *Future Generation Computer Systems*, *150*, 1–9.

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, *106*, Article 107404.

Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). BASNet: Boundary-aware salient object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7471–7481.

Ramacharya, B. P., Patil, M. R., & Keralkar, S. (2023). Fast partial image encryption with fuzzy logic and chaotic mapping. *Evolutionary Intelligence*, 667–683.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. ArXiv, arXiv:1804.02767.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Sheela, S. J., & Suresh, K. V. (2024). Real time region of interest based chaotic image cryptosystem for IoT applications. *Multimedia Tools and Applications*, *83*, 16161–16177.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Singh, K. N., Baranwal, N., Singh, O. P., & Singh, A. K. (2024). DeepENC: Deep learning-based ROI selection for encryption of medical images through key generation with multimodal information fusion. *IEEE Transactions on Consumer Electronics*, 1.

Singh, P., Devi, K. J., Thakkar, H. K., & Kotecha, K. (2022). Region-based hybrid medical image watermarking scheme for robust and secured transmission in IoMT. *IEEE Access*, *10*, 8974–8993.

Singh, K. N., Singh, O. P., Baranwal, N., & Singh, A. K. (2022). An efficient chaos-based image encryption algorithm using real-time object detection for smart city applications. *Sustainable Energy Technologies and Assessments*, *53*, Article 102566.

Song, W., Fu, C., Lin, Z., Zhang, Y., Chen, J., & Sham, C.-W. (2024). Batch medical image encryption using 3D latin cube-based simultaneous permutation and diffusion. *Signal, Image and Video Processing*, *18*(3), 2499–2508.

Song, W., Fu, C., Zheng, Y., Cao, L., Tie, M., & Sham, C.-W. (2022). Protection of image ROI using chaos-based encryption and DCNN-based object detection. *Neural Computing and Applications*, *34*, 5743–5756.

Song, W., Fu, C., Zheng, Y., Tie, M., Liu, J., & Chen, J. (2023). A parallel image encryption algorithm using intra bitplane scrambling. *Mathematics and Computers in Simulation*, *204*, 71–88.

Song, W., Fu, C., Zheng, Y., Zhang, Y., Chen, J., & Wang, P. (2024). Batch image encryption using cross image permutation and diffusion. *Journal of Information Security and Applications*, *80*, Article 103686.

Su, Y., Teng, L., Liu, P., Unar, S., Wang, X., & Fu, X. (2023). Visualized multiple image selection encryption based on log chaos system and multilayer cellular automata saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*, (9), 4689–4702.

Thodi, D. M., & Rodríguez, J. J. (2007). Expansion embedding techniques for reversible watermarking. *IEEE Transactions on Image Processing*, *16*(3), 721–730.

Tsai, P., Hu, Y.-C., & Yeh, H.-L. (2009). Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Processing*, *89*(6), 1129–1143.

Wang, L., Chen, Z., Sun, X., & He, C. (2023). Color image ROI encryption algorithm based on a novel 4D hyperchaotic system. *Physica Scripta*, *99*(1), Article 015229.

Wang, L., Chen, Z., Sun, X., & He, C. (2025). Region of interest encryption algorithm for images based on lifting scheme and object detection. *Cluster Computing*, *28*(1), 8.

Wang, M., Konrad, J., Ishwar, P., Jing, K., & Rowley, H. (2011). Image saliency: From intrinsic to extrinsic context. In *CVPR 2011* (pp. 417–424).

Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H., & Yang, R. (2022). Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 3239–3259.

Wang, J., Liu, L., Xu, M., & Li, X. (2022). A novel content-selected image encryption algorithm based on the ls chaotic model. *Journal of King Saud University - Computer and Information Sciences*, *34*(10, Part A), 8245–8259.

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., & Ruan, X. (2017). Learning to detect salient objects with image-level supervision. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 3796–3805).

Wang, J., Zhang, R., & Liu, J. (2024). Partial-privacy image encryption algorithm based on time-varying delayed exponentially controlled chaotic system. *Nonlinear Dynamics*, *112*, 10633–10659.

Wei, C., & Li, G. (2022). A selective image encryption scheme using LICC hyperchaotic system. *IET Image Processing*, *16*, 3342–3358.

Wu, R., Feng, M., Guan, W., Wang, D., Lu, H., & Ding, E. (2019). A mutual learning method for salient object detection with intertwined multi-supervision. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8142–8151.

Wu, Y.-H., Liu, Y., Xu, J., Bian, J.-W., Gu, Y.-C., & Cheng, M.-M. (2022). MobileSal: Extremely efficient RGB-D salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(12), 10261–10269.

Wu, Y.-H., Liu, Y., Zhang, L., Cheng, M.-M., & Ren, B. (2022). EDN: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing*, *31*, 3125–3136.

Ye, G., & Guo, L. (2024). A visual meaningful encryption and hiding algorithm for multiple images. *Nonlinear Dynamics*, *112*(16), 14593–14616.

Zhang, J., Guo, J., & Lu, D. (2023). An efficient image encryption algorithm based on S-box and DNA code. *Measurement: Sensors*, *29*, Article 100894.

Zhang, H., Kone, M. M. K., Ma, X.-Q., & Zhou, N.-R. (2025). Frequency-domain attention-guided adaptive robust watermarking model. *Journal of the Franklin Institute*, *362*(3), Article 107511.

Zhao, X., Pang, Y., Zhang, L., Lu, H., & Zhang, L. (2020). Suppress and balance: A simple gated network for salient object detection. *Computer Vision – ECCV 2020*, 35–51.

Zhao, T., & Wu, X. (2019). Pyramid feature attention network for saliency detection. In *2019 IEEE/CVF conference on computer vision and pattern recognition* (pp. 3080–3089).

Zheng, J., Yang, L., Li, Y., Yang, K., Wang, Z., & Zhou, J. (2023). Lightweight vision transformer with spatial and channel enhanced self-attention. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1492–1496).

Zhou, N.-R., Hu, L.-L., Huang, Z.-W., Wang, M.-M., & Luo, G.-S. (2024). Novel multiple color images encryption and decryption scheme based on a bit-level extension algorithm. *Expert Systems with Applications*, *238*, Article 122052.

Zhou, S., Qiu, Y., Qi, G., & Zhang, Y. (2023). A new conservative chaotic system and its application in image encryption. *Chaos, Solitons & Fractals*, *175*, Article 113909.

Zou, J.-Z., Chen, M.-X., & Gong, L.-H. (2025). Invisible and robust watermarking model based on hierarchical residual fusion multi-scale convolution. *Neurocomputing*, *614*, Article 128834.

Zou, C., Liu, Y., Yang, Y., Zhou, C., Yu, Y., & Shang, Y. (2025). A privacy-preserving license plate encryption scheme based on an improved YOLOv8 image recognition algorithm. *Signal Processing*, *230*, Article 109811.