applied sciences

MDPI

Article

A High-Capacity and High-Security Image Steganography Network Based on Chaotic Mapping and Generative Adversarial Networks

Lin Huo [1,*], Ruipei Chen [1], Jie Wei [2] and Lang Huang [1]

[1] School of Computer and Electronic Information, Guangxi University, Nanning 530004, China; baconcrp@163.com (R.C.); 2213301017@st.gxu.edu.cn (L.H.)
[2] Guangxi Key Laboratory of Digital Infrastructure, Guangxi Information Center, Nanning 530000, China
* Correspondence: lhuo@gxu.edu.cn

Abstract: With the enhancement of information volume, people are not satisfied with transmitting only a single secret image at a time but chase to hide multiple secret images in a single picture; however, the large-capacity steganographic scale can easily lead to the degradation of the quality of the image, which attracts the attention of eavesdroppers. In this paper, we propose a ***C***haotic mapping-en***H***anced im***A***ge **S**teganography n***E***twork (CHASE), which pioneers to hide colour images in grey images and reduces the difference between the container image and the cover image through the image permutation method, so as to enhance the security of the steganography. The method demonstrates excellent steganalysis resistance in experiments and introduces Generative Adversarial Networks (GANs) to improve the image fidelity in large-capacity steganographic scales. The fusion of chaotic mapping and GAN optimisation enables the steganographic network to simultaneously balance security and image quality. The experimental results show that CHASE can keep the secret image with good invisibility under large-capacity steganographic scales, and at the same time, it can reveal the secret image with high fidelity, and its steganalysis-resistant capability is much better than other state-of-the-art methods.

Keywords: image steganography; chaotic mapping; generative adversarial network; invertible neural networks; anti-steganalysis

Citation: Huo, L.; Chen, R.; Wei, J.; Huang, L. A High-Capacity and High-Security Image Steganography Network Based on Chaotic Mapping and Generative Adversarial Networks. *Appl. Sci.* **2024**, *14*, 1225. https://doi.org/10.3390/app14031225

Academic Editors: George Drosatos, Konstantinos Rantos and Konstantinos Demertzis

Received: 17 December 2023
Revised: 26 January 2024
Accepted: 26 January 2024
Published: 1 February 2024

1. Introduction

In order to ensure the security of transmitted information, people often use cryptographic methods to encrypt secret information, thus hiding the meaning of the information and making the information unreadable, but this practice cannot hide the existence of the information. Irregular ciphertext information is often more susceptible to the attacker's suspicion during the transmission process, which generates a new security problem [1] of how to secretly transmit information under the eavesdropping of the attacker. Information-hiding technology may be the answer to this problem. This technology hides secret information in multimedia carriers (e.g., images, text, etc.) and then saves or forwards it, which does not easily draw attention, and only the receiver can extract the secret information, thus realising the purpose of privacy data preservation and sharing [2]. Thanks to the extensive use of images in practice, image steganography has gradually developed to be a popular field of information hiding [3].

In image steganography, a cover/host image is used as a container for secret information, and the resulting image containing the secret information is called a stego/container image. Image steganography needs to successfully hide secret information while minimising the impact on the image quality to avoid significant impact on visual perception. Secondly, the capacity limitation of steganography is an important consideration, as people always chase to hide more secret information. In addition, steganography needs to be

secure, i.e., the presence of secret information is not detected when an attacker uses a steganalysis tool.

Traditional image steganography typically involves embedding secret information by modifying pixel values in the spatial domain [4,5] or spectral components in the transform domain [6–8]. However, these methods often degrade the quality of the stego image and introduce noticeable changes to its statistical properties, making them susceptible to steganalysis detection. With advancements in deep learning technology, new approaches to image steganography have emerged. For instance, the SGAN steganographic network [9] utilizes a deep convolutional neural adversarial network (DCGAN) to generate cover images that better align with real distribution, enhancing visual consistency for embedded secrets. Some methods [10–12] leverage Adversarial Neural Networks to address the statistical alteration issue present in traditional image steganography. These advancements mark a fusion of image steganography with deep learning techniques, presenting innovative solutions to improve both security and visual perception.

Encoding and decoding operations in encoder-decoder networks exhibit similarity to the hiding and reconstruction processes in information hiding. To leverage this similarity, Hayes et al. [13] proposed the SteGAN text steganography model, which autonomously learns the hiding and reconstruction processes of secret information, significantly enhancing steganographic capacity. Wang et al. [14] improved SteGAN further, enhancing the realism of the stego image. However, both methods are limited to hiding binary sequence information. To enhance steganographic capacity further, Baluja et al. [15] proposed the Deep Steganography model, achieving the steganographic effect of hiding one image within another, and later improving the model to enable large-capacity image steganography—hiding two images in a single image [16]. Despite these advancements, these methods, with separate designs for the encoder and decoder, involve irreversible processes for hiding and reconstructing information, leading to potential image quality degradation and security vulnerabilities against steganalysis.

This paper aims to integrate the strengths of image steganography based on encoder-decoder networks and generative adversarial networks, proposing the CHASE image steganography network to achieve high capacity, invisibility, and security. Diverging from previous designs, we incorporate the invertible neural network (INN) for image encoding and decoding, treating image hiding as a forward process and secret information reconstruction as a backward process within a single trained network. Additionally, we introduce a novel chaotic mapping permutation algorithm to enhance security. Through extensive experiments, we validate the superior performance of our network in image steganography tasks. Key contributions include:

- Designing CHASE, a novel invertible high-capacity image steganography network capable of hiding a multi-channel colour secret image within a single-channel grey cover image in a single steganography process.
- Proposing an image permutation algorithm based on Logistic chaotic mapping, utilising encrypted secret images in the steganography process to enhance security.
- Combining encoder-decoder steganography structure and the adversarial learning concept of generative adversarial networks to improve image quality in stego and revealed secret images at large-capacity scales, with generated stego images exhibiting high resistance to steganalysis.

The subsequent sections are structured as follows: Section 2 provides an overview of relevant research and existing methodologies in image steganography and INN. Section 3 comprehensively outlines our proposed image steganography network, based on the INN and GAN, elucidating the implementation of the image permutation algorithm. Section 4 presents and scrutinizes the experimental outcomes. Finally, Section 5 encapsulates the paper's findings and offers a forward-looking perspective.

2. Related Work

Image steganography stands as a prominent research domain within information security. In this context, we succinctly examine seminal contributions in image steganography alongside recent advancements in invertible neural networks.

2.1. Image Steganography

Traditional image steganography is broadly categorized into spatial and transform domains. Spatial techniques, like the LSB algorithm, are simple but often result in noticeable quality degradation and artifacts. Transform domain methods, involving DFT, DWT, and DCT, aim to embed data into spectral components but may overlook natural pixel distribution, impacting image quality. Researchers address this by exploring steganography in complex image texture regions, minimizing steganographic distortion to embed pixels judiciously. Approaches like S-UNIWARD [17], WOW [18], and HUGO [19] adopt this concept. However, these methods still rely on human-designed processes, making them vulnerable to steganalysis attacks.

The integration of deep learning into image steganography has spurred the development of various methods, broadly categorized into three classes based on the embedding process: (1) Enhanced Cover Image Generation; (2) Steganographic Distortion Framework Design; and (3) Stego Image Generation.

In the class of enhanced cover image generation, most methods utilize a Generative Adversarial Network (GAN) to create cover images with intricate textures suitable for steganography. Volkhonskiy et al. [9] were pioneers in using GANs for image steganography, employing the generator and discriminator for enhanced cover image generation. Shi et al. [20] enhanced this method by incorporating the Wasserstein distance as the loss function and modifying the generative network structure, improving the quality of the generated images. However, this class faces challenges in reduced steganographic capacity, and the potential for suspicion due to unrealistic cover images is a concern for attackers. The steganographic distortion framework class is centred on employing neural networks to formulate more rational distortion functions. Through adversarial training, the neural network learns the embedded distortion probability of each pixel, thereby guiding the modification of the cover image. Tang et al. [10] introduced a GAN to automatically learn steganographic distortion in images. This method proved effective in comparison with human-designed adaptive steganography algorithms and demonstrated notable resistance against steganalysis attacks.

The category of stego image generation integrates encoder-decoder networks into the realm of image steganography, showcasing continued advancements. Hayes et al. [13] introduced the HayesGAN steganographic network, pioneering the concealment of text information within images using coding networks. This method involves inputting text information and cover images into the encoder network to obtain stego images, which are then reconstructed by the decoder network. To enhance security, the stego image and cover image undergo steganography analysis and detection through a discriminative network. Zhang et al. [21] significantly increased the capacity for hidden text information with the SteganoGAN, which, achieving a large-capacity image steganography of 4.4 bpp, this network produces more realistic stego images.

Baluja [15], employing DCGAN, achieved steganography for colour image of the same size. The network structure comprises three components: a pre-processing network resizing the secret image to match the cover image's size, an encoder network combining the secret image and cover image to generate the stego image, and a decoder network facilitating the extraction of the secret image. Rehman et al. [22] made improvements on this foundation, but both steganographic networks led to the issue of colour distortion. Zhang et al. [23] addressed this by converting the RGB format of the steganography-performed image to YCrYb format. Since the Y channel exclusively retained semantic information without colour details, the secret grey image was hiding in the Y-channel of cover image. Subsequently, the obtained stego image was combined with the Cr and Cb channel images,

effectively resolving the colour distortion problem. However, this approach also resulted in a two-thirds reduction in the steganographic capacity. The inherent link between the concepts of encoder-decoder networks and concealment-extraction has garnered significant attention in image steganography methods. However, the irreversibility in the design of these methods using encoder-decoder networks has led to a certain level of degradation in secret information extraction accuracy, consequently limiting improvements in steganographic capacity.

2.2. Invertible Neural Network

The flow model entails the mapping of a simple a priori distribution, typically Gaussian, to the true distribution of the data through a series of invertible transformations. This mapping is invertible, allowing for both the generation of samples from the data (forward flow) and the reconstruction of the data from the samples (backward flow). The earliest flow model, NICE, was introduced by Dinh et al. [24]. It comprises multiple additive coupling layers, significantly enhancing the capacity to learn nonlinear deterministic transformations of data through the scaling transformation layer. Subsequently, Dinh et al. [25] extended NICE by introducing a convolutional neural network (CNN), resulting in RealNVP. They improved the additive coupling layer to an affine coupling layer, making it suitable for image processing tasks. Kingma et al. [26] further advanced the field by introducing invertible 1×1 convolution and proposing Glow, a model capable of synthesizing images with enhanced perceptual fidelity.

Invertible neural networks serve as a practical realization of the flow model, garnering significant interest from researchers in the imaging domain. Within the realm of super-resolution, Lugmayr et al. [27] presented SRFlow, a normalized flow-based super-resolution method that effectively learns the conditional distribution of high-resolution outputs given low-resolution inputs, resulting in high-quality images. Ardizzone et al. [28] introduced conditional invertible neural network (cINN), showcasing an architecture capable of efficiently pre-processing conditional inputs into valuable features and generating diverse samples with sharper images. In the context of medical image reconstruction, Denker et al. [29] introduced the conditional flow model cINN, enhancing reconstruction quality by substituting the standard Gaussian distribution with a radial Gaussian distribution. Addressing the rendering of visually appealing sRGB images, Xing et al. [30] devised an invertible image signal processing (InvISP) pipeline, demonstrating improved quality in both rendered sRGB images and reconstructed RAW data. For image rescaling, Xiao et al. [31] developed an invertible rescaling network (IRN) capable of producing visually superior low-resolution images through the inverse bijection process, while also reconstructing high-resolution images with quality closely resembling the original image. Invertible neural networks have found application in conjunction with image steganography. Lu et al. [32] introduced an invertible steganographic network (ISN) designed to conceal multiple colour images within one colour image, exhibiting enhanced steganographic capacity while maintaining good invisibility. Meanwhile, Jing et al. [33] proposed the HiNet steganography framework, incorporating a low-frequency wavelet loss to enhance security while preserving high image quality. Building upon HiNet, Guan et al. [34] introduced a focus map mapping module to guide the location of image steganography effectively, thereby improving overall image hiding capacity.

3. Proposed Approach

3.1. Overview

Figure 1 illustrates our comprehensive steganography model. The primary objective of the proposed steganography in this paper was to improve the hiding capacity of image steganography while maintaining high-fidelity image quality. Additionally, we aimed to bolster steganography's resilience against steganalysis attacks by incorporating an image permutation algorithm based on Logistic chaotic mapping. In this section, we begin by delineating the concealed secret information's location and presenting an image

permutation algorithm grounded in chaotic mapping. Subsequently, we provide a detailed analysis of our image steganography network, CHASE, treating its encoding (hiding) and decoding (reconstructing) processes as the network's forward and backward processes, respectively. We also introduce an adversarial neural network for adversarial training to enhance image quality. Finally, we expound on the network's training strategy and elucidate the rationale behind constructing the loss function.

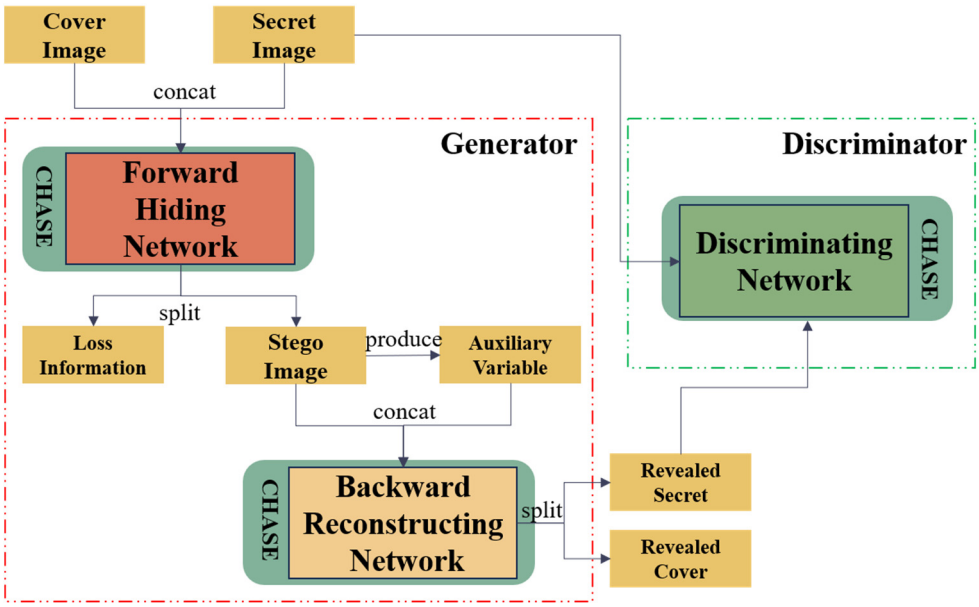

Figure 1. The process in which the cover image and secret image undergo the forward hiding network, resulting in the generated stego image and associated loss information. Subsequently, the stego image, along with auxiliary variables produced from stego image, is input into the backward reconstructing network. This network reveals the cover image and secret image. The revealed secret image, alongside the original secret image, undergoes assessment by the discriminator for differentiation. Through iterative adversarial training, the quality of the reconstructed secret image is refined.

3.2. Network Architecture

3.2.1. Steganography Position

In contrast to conventional image steganography methods that conceal information within RGB or grey image, our approach in this paper involves hiding secret information within the Y channel of YCrCb images. Some RGB-based steganography techniques tend to distort the colour of the stego images. This distortion arises from the fact that RGB images consist of three channels (R, G, B), encompassing semantic, luminance, and colour information. When concealing secret information, the distribution of this information is inevitably disrupted, resulting in colour accuracy issues.

$$\begin{cases} Y = 0.299 * R + 0.587 * G + 0.114 * B \\ Cr = 0.5 * (R - Y) * 0.713 + 128 \\ Cb = 0.5 * (B - Y) * 0.564 + 128 \end{cases} \tag{1}$$

$$\begin{cases} R = Y + 1.403 * (Cr - 128) \\ G = Y - 0.714 * (Cr - 128) - 0.344 * (Cb - 128) \\ B = Y + 1.773 * (Cb - 128) \end{cases} \tag{2}$$

As depicted in Figure 2, within the YCrCb encoding format, the Cr and Cb channels encapsulate both partial semantic and complete colour information of the image. Meanwhile, the Y channel contains only partial semantic information and luminance information. By concealing information solely in the Y channel, we mitigated the problem of colour

distortion during embedding. The conversion between YCrCb and RGB coding formats is described by Equations (1) and (2). Unlike Zhang et al. [23], who only hides grey image in the *Y* channel, our approach allows for hiding colour image in *Y* channel, significantly enhancing steganographic capacity.

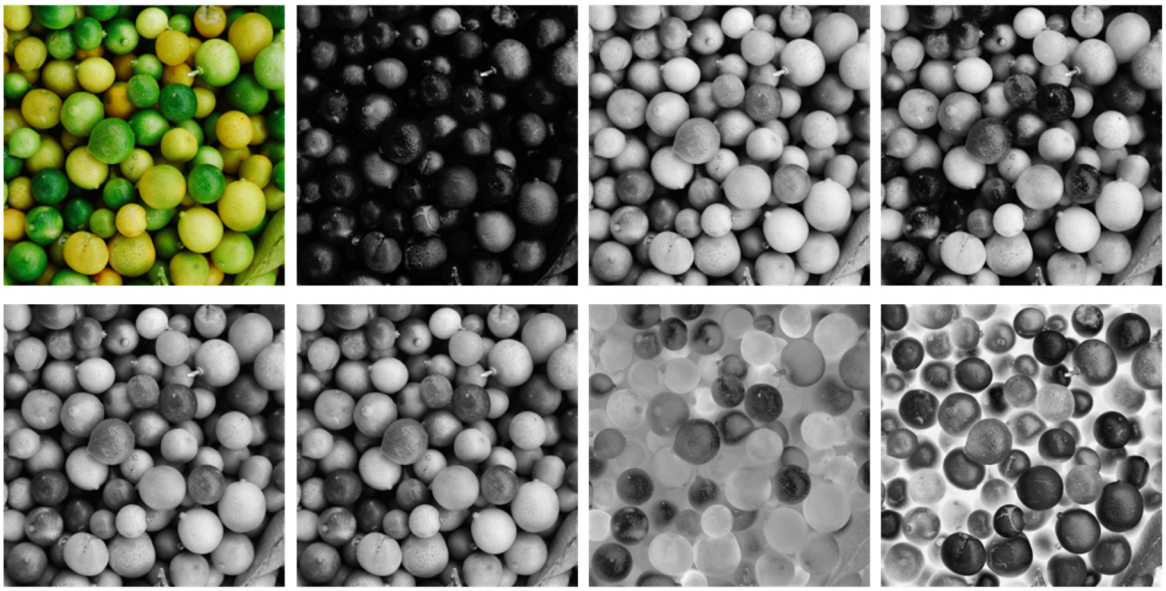

Figure 2. In the first row, the first column of images is the original RGB image, and the second, third, and fourth columns are, in order, images with *R*, *G*, and *B* channels. And in the second row, the first column image is the greyscale image of the original RGB image, and the second, third, and fourth columns are the images of *Y*, *Cr*, and *Cb* channels in turn.

3.2.2. Image Permutation

During information hiding, we found that the presence of a large, solid colour area in the chosen cover image or complex texture in the secret image can lead to the error image, i.e., the residual image between the stego image and the cover image, containing discernible texture information from the secret image. This poses a significant security risk if the set of cover images used for training is inadvertently disclosed. To address this concern, we integrated image permutation with image steganography to minimize the distinction between the stego image and the cover image. This ensures that even if the image set is exposed, attackers cannot extract valuable information from the error image. Leveraging the sensitivity of chaotic systems to initial values and their ability to generate variable and intricate pseudo-random sequences [35], we proposed an image permutation algorithm based on Logistic chaotic mapping. The permutation process, depicted in Figure 3, involves four primary steps, and its reversal process serves as the inverse of the scrambling process.

Figure 3. The first image represents the original image, the second image is generated by traversing the original image according to a specific rule, and subsequent permutation produces the third image. Without the correct key and the inverse permutation algorithm, an incorrectly restored image (fourth image) is obtained.

Step 1: Pixel Position Choosing

Iterate over image rows and columns, selecting pixel values alternatively at odd and even positions for each row and column. This process creates four sub-images with pixel positions denoted as (odd, odd), (odd, even), (even, odd), and (even, even). The dimensions of these sub-images are one-quarter of the original image.

Step 2: Chaotic Sequence Generation

The Logistic chaotic mapping is expressed in Equation (3), where the control parameter μ ranges between (0, 4], but when the value range is within the interval of (3.5699456, 4], it enters the chaotic state [36]. We utilized the logistic chaotic map twice with different parameters to generate random sequences. As we can see in Figure 4, initially, using the Logistic chaotic mapping (F_Logistic) to generate a chaotic sequence (F_Sequence) of the same size as the sub-image for reordering, the position index of each element in this ordered sequence is calculated based on the original chaotic sequence. Subsequently, we used Logistic chaotic mapping (S_Logistic) again with distinct control parameters and initial values to generate four copies of the sequence (S_Sequence) for reordering, each having a quarter of the length of F_Sequence. Calculations are performed to obtain the position index of each element in the ordered sequence, guiding the reordering of F_Sequence sequentially to intensify its disorder. Finally, Guide_Index is acquired and utilized as guidance to determine the pixel positions of the sub-image to be scrambled.

$$X_{n+1} = \mu \times X_n \times (1 - X_n), \quad n = 0, 1, 2 \ldots \tag{3}$$

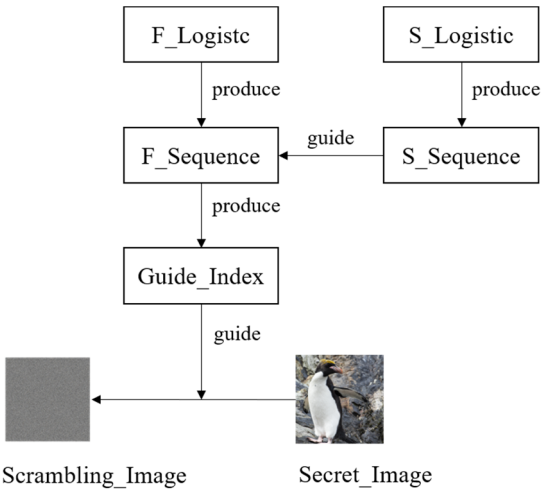

Figure 4. Utilizing two chaotic mappings, the resulting chaotic sequence is further scrambled. This final disrupted sequence is then employed to guide the permutation of the image, thereby enhancing the overall image scrambling.

Step 3: Iterative Scrambling of Sub-Images

Continuously repeat Steps 2 and 3 until all four sub-images have undergone scrambling.

Step 4: Random Exchange of Sub-Images

Perform a random exchange of positions for the scrambled sub-images. To maintain consistency in size with the cover image, concatenate these sub-images back into the original image based on their row and column positions. Output the final scrambled image after this exchange.

This algorithm iteratively scrambles sub-images in a chaotic manner, guiding the disordering of pixel positions. After the final iteration, a random exchange of sub-image positions is performed, resulting in the output of the scrambled image. The traversal of sub-images and the transformation of the final position of the sub-images are conducted to

enhance the efficacy of image permutation. Improving the degree of scrambling contributes to enhanced security.

3.2.3. Chaos Mapping Enhanced Image Steganography Network (CHASE)

Information-hiding encoder-decoder networks proposed by researchers such as Baluja [15] and Rehman et al. [22] typically adopt an independent design for encoder and decoder networks. This approach results in the irreversibility of the secret information encoding and decoding processes, consequently diminishing the success rate of secret information restoration. Taking inspiration from the architecture of invertible neural networks like Dihn [25] and Lu [32], our proposed CHASE model integrates the encoding and decoding of secret information within the same network. Figure 5 illustrates our steganographic network (CHASE), and Table 1 provides an explanation of the notation used below.

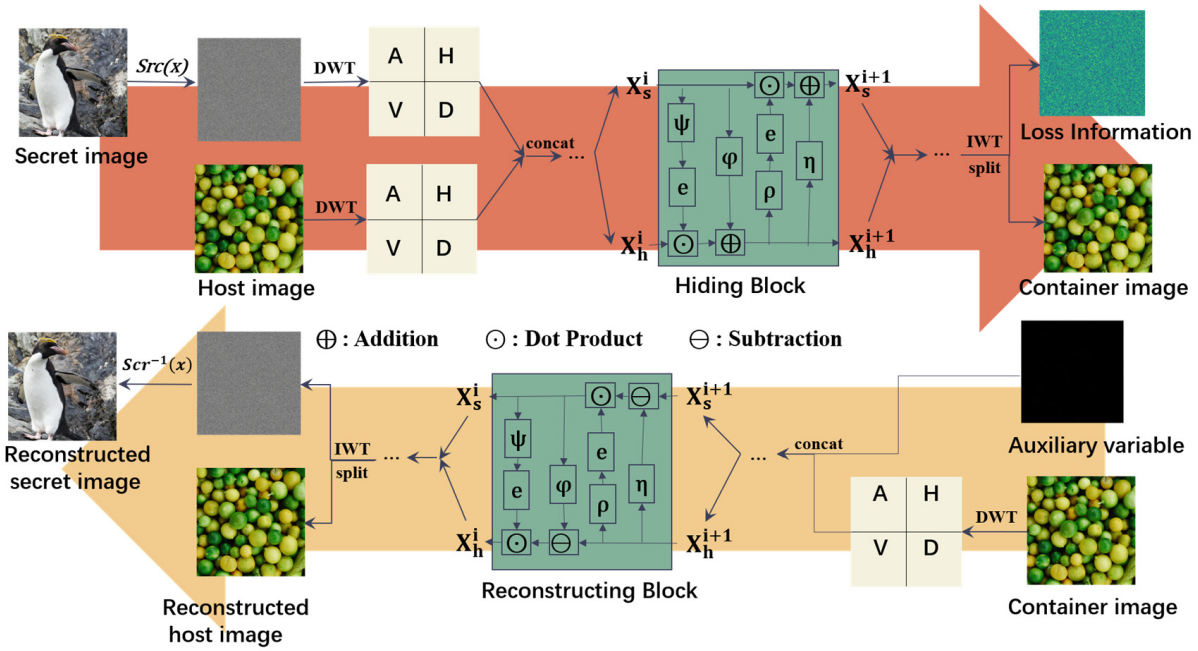

Figure 5. The framework of CHASE. During forward hiding, the Haar wavelet transforms the scrambled secret image and cover image. Invertible blocks then process the transformed data, generating the stego image and loss information. In backward reconstruction, auxiliary variable and the stego image pass through invertible blocks to reconstruct the cover image and scrambled secret image. The inverse scrambling of the latter yields the final secret image. In the hiding and reconstructing blocks, ρ, η, φ, and ψ can be arbitrary functions, and we utilized Dense block [37] to represent them to enhance the image processing effect.

Table 1. Summary of notations used in this article.

Notation	Description
X_h	Cover image
X_s	Secret image
X_c	Stego image
$Scr(x)/Scr^{-1}(x)$	Image scrambling process/Image inverse scrambling process
$F(x)/F^{-1}(x)$	Hiding process/Reconstruction process
$\hat{X}_s$	Reconstructed secret image
$\overline{\overline{X_s}}$	Inverse Scrambling reconstructed secret image
$\hat{X}_h$	Reconstructed host image
r	Loss function generated during forward hiding process
z	Auxiliary variable used to assist in reconstructing images

In the forward hiding process of the network, X_h is transformed from the RGB encoding format to the YcrCb format, with the Y channel selected as the actual cover image. Subsequently, the Y channel image and X_s undergo separate transformations using the Haar wavelet, a discrete wavelet transform variant that is easy to implement. Hiding information in the frequency domain is better advantageous for preserving the visual quality of the image than in the spatial domain. Additionally, the Haar wavelet transform is orthogonal, ensuring the retention of image information, and enabling complete reconstruction of the original image during the inverse transformation. The two transformed images are input into the network.

Following the invertible block sequence and, ultimately, the inverse wavelet transform (IWT), the Y channel image with intact information and the loss information r are obtained. The Y channel image at this stage undergoes further transformation to yield X_c, which is utilized for actual transmission. In the backward process, the Y channel image is initially extracted from X_c. The reconstructed Y channel image and the scrambled secret image $\hat{X}_s$ then undergo the scrambling inverse process to ultimately obtain the reconstructed secret image $\overline{\overline{X_s}}$. We designed the hiding and reconstructing processes as reciprocal problems, and this process can be expressed by the following formula:

$$IWT(F(DWT(X_h), DWT(Src(X_s)))) = (X_c, r) \tag{4}$$

$$IWT\left(F^{-1}(DWT(X_c), z)\right) = \left(\hat{X}_h, Scr^{-1}(\hat{X}_s) = \overline{\overline{X_s}}\right) \tag{5}$$

Invertible Block. The designed hiding and reconstruction networks share identical sub-modules and network parameters, employing the affine coupling layer as the invertible block. In contrast to the additive coupling layer, the affine coupling layer adeptly captures intricate relationships in input information and effectively manages high-dimensional data distribution. The input accepted at the beginning of the reversible block is (X_h, X_s), which can also be denoted as (X_h^1, X_s^1) to indicate the first input. Throughout the forward hiding process, multiple invertible blocks process information. For the i-th invertible block, its input (X_h^i, X_s^i) and output (X_h^{i+1}, X_s^{i+1}) can be represented by the following formula:

$$\begin{cases} X_h^{i+1} = e\left(\alpha \odot \psi\left(X_s^i\right)\right) \odot X_h^i \oplus \varphi\left(X_s^i\right) \\ X_s^{i+1} = e\left(\alpha \odot \rho\left(X_h^{i+1}\right)\right) \odot X_s^i \oplus \eta\left(X_h^{i+1}\right) \end{cases} \tag{6}$$

where $\rho(\cdot)$, $\eta(\cdot)$, $\varphi(\cdot)$, and $\psi(\cdot)$ can represent arbitrary functions. Here, $e(\cdot)$ signifies a Sigmoid function, and the coefficient α acts as a regularization factor to mitigate numerical instability arising from the $e(\cdot)$ function. Following the last forward hiding block, the outcomes undergo IWT to obtain the stego image and loss information, respectively. For the backward reconstruction module, the information flow direction is inverted. In the i-th module, the output $(\hat{X}_h^i, \hat{X}_s^i)$ from the i + 1-th module serves as input $(\hat{X}_h^{i+1}, \hat{X}_s^{i+1})$, as expressed by the following equation:

$$\begin{cases} \hat{X}_s^i = \left(\hat{X}_s^{i+1} \ominus \eta\left(\hat{X}_h^{i+1}\right) \odot \exp\left(-\alpha \odot \rho\left(\hat{X}_h^{i+1}\right)\right)\right) \\ \hat{X}_h^i = \left(\hat{X}_h^{i+1} \ominus \varphi\left(\hat{X}_s^i\right)\right) \odot \exp\left(-\alpha \odot \psi\left(\hat{X}_s^i\right)\right) \end{cases} \tag{7}$$

Initially, the inverse extraction model takes the stego image and auxiliary information as input. At the final stage, executing IWT and image inverse permutation yields the reconstructed secret image.

The Loss information r and auxiliary variable z. In the forward hiding process, the loss of information in the cover and secret image results in the generation of loss information r in the output. For successful secret information reconstruction in the backward reconstruction module, both the loss information r and the stego image must be used as input. In actual transmission processes, transmitting loss information is impractical, as it may make it easier for an attacker to deduce the original secret information or result in

information leakage. Moreover, transmitting loss information poses a risk of information leakage. While some steganography methods employing reversible neural networks utilize auxiliary variables sampled from standard Gaussian distributions during the reverse extraction process, this approach may compromise the accuracy of secret image extraction. The success of deep learning-based image steganography is attributed to the frequency differences between the cover image and the secret image [38]. To enhance secret image extraction accuracy, we leveraged the high-frequency sub-bands of the Stego image as auxiliary variables, containing partial information of the secret image to better assist in the reverse extraction operation.

Generative Adversarial Networks. The introduction of the auxiliary variable z facilitates secret information extraction but may compromise the completeness of the secret information, leading to a reduction in the quality of the secret image. To enhance the fidelity of the reconstructed secret image, we incorporated a Generative Adversarial Network (GAN) [39]. The GAN framework comprises a generator responsible for image generation and a discriminator tasked with evaluating the difference between the generated and real images.

In our setup, CHASE serves as the generator to generate the reconstructed secret image and the original secret image as the real image. The generator aims to produce a reconstructed image indistinguishable from the real image, while the discriminator is trained to differentiate between the generated and real images. The discriminator network structure is detailed in Table 2. Through iterative adversarial training, the distribution difference between the reconstructed secret image generated by the backward reconstruction process using auxiliary variable z and the distribution of the original secret image diminishes.

Table 2. The structure of the discriminator network. It employs convolution kernels of various scales to perform dimensionality reduction on the input, effectively extracting meaningful features. The CBAM [40] self-attention mechanism was employed to enhance the representation of these features.

Layers	Process	Output Size
Input	/	$3 \times 256 \times 256$
Layer 1	3×3 Conv + LeakyReLu	$64 \times 256 \times 256$
Layer 2	4×4 Conv + BatchNorm + LeakyReLu	$64 \times 128 \times 128$
Layer 3	3×3 Conv + BatchNorm + LeakyReLu	$128 \times 128 \times 128$
Layer 4	4×4 Conv + BatchNorm + LeakyReLu	$128 \times 64 \times 64$
Layer 5	3×3 Conv + BatchNorm + LeakyReLu	$256 \times 64 \times 64$
Layer 6	4×4 Conv + BatchNorm + LeakyReLu	$256 \times 32 \times 32$
Layer 7	3×3 Conv + BatchNorm + LeakyReLu	$512 \times 32 \times 32$
Layer 8	4×4 Conv + BatchNorm + LeakyReLu	$512 \times 16 \times 16$
Layer 9	3×3 Conv + BatchNorm + LeakyReLu	$512 \times 16 \times 16$
Layer 10	4×4 Conv + BatchNorm + LeakyReLu	$512 \times 8 \times 8$
Layer 11	CBAM	$1 \times 32{,}768$
Layer 12	FC	1×100
Output	FC	1×1

The introduction of the GAN aims to enhance the quality of the reconstructed image, bringing it closer to the original image. Continuous adversarial training within the backward reconstruction process enables the reconstruction of higher-quality secret images using the auxiliary variable z as input. The primary objective of this deep learning framework is to achieve superior visual quality and fidelity while preserving secrecy.

3.3. Optimization Strategy

Model Training Strategy. Due to the potential instability in the training process of Generative Adversarial Network (GAN), a two-stage training approach was employed to enhance stability and improve the quality of generated images. In the initial phase, training network operates without GAN, which only use z to help reconstruct secret image. Once a relatively stable state is reached in the quality of the generated loss information r and the

secret image reconstructed by the auxiliary variable z, the training proceeds to the second phase. In the second phase, a discriminator network is utilized to train the reconstructed secret image to distinguish it from the original secret image. This aims to ensure that the distribution of secret images reconstructed by CHASE aligns with the distribution of the original secret images, thereby enhancing the quality and fidelity of the generated images. The two-stage training strategy is implemented to address potential challenges in GAN training, promoting stability and aligning the generated images more closely with the desired distributional properties.

Loss Function. To ensure the fidelity of both the stego and reconstructed secret images to the original cover image and secret image, respectively, two crucial losses are employed throughout the training process:

- Hiding Loss. In the forward hiding process, the network conceals the secret information within the cover image to generate the stego image. The objective is to make the stego image visually close to the cover image. Hence, the hiding loss is defined as follows, where N represents the number of reversible blocks:

$$L_{hid}(\Theta) = \sum_{n=1}^{N} \ell_{\text{hid}}\left(X_h^{(n)}, X_c^{(n)}\right) \tag{8}$$

- Reconstruction Loss. The secret image reconstructed by the backward reconstruction process should be kept consistent with the original secret image, and for this purpose, the reconstruction loss is defined in the following form, where N represents the number of reversible blocks:

$$L_{rec}(\Theta) = \sum_{n=1}^{N} \ell_{\text{rec}}\left(X_s^{(n)}, \overline{\overline{X}}_s^{(n)}\right) \tag{9}$$

And the next two loss functions are used in each of the two stages of training.

- Loss information r loss. In stage 1, we performed $L2$ regularization of the loss information r as a distribution loss function to constrain the loss information distribution to be more concentrated around values close to zero, thus reducing the complexity of the model, making the model smoother and the training more stable.

$$L_{lin} = \|r\|_2^2 \tag{10}$$

So, the total loss function at stage 1 is expressed as:

$$L_{s1} = \lambda_h L_{hid} + \lambda_r L_{rec} + \lambda_l L_{lin} \tag{11}$$

- GAN Loss. In Stage 2, the cross-entropy loss function is employed as the distribution loss to quantify the disparity between the distribution of the reconstructed secret image and the original secret image. By considering the original secret image as the ground truth and the reconstructed secret image as the predicted distribution in the GAN model, minimizing the cross-entropy loss encourages the model to align its predicted distribution closely with the ground truth distribution. The GAN loss is defined in the following form, where N represents the number of reversible blocks.

$$L_{gan}(\Theta) = -\sum_{n=1}^{N}\left[X_s^{(n)} log\overline{\overline{X}}_s^{(n)} + \left(1 - X_s^{(n)}\right)\log\left(1 - \overline{\overline{X}}_s^{(n)}\right)\right] \tag{12}$$

Therefore, the total loss function at stage 2 is expressed as:

$$L_{s2} = \lambda_h L_{hid} + \lambda_r L_{rec} + \lambda_g L_{gan} \tag{13}$$

4. Experimental Results

4.1. Implementation and Setup Details

Datasets and settings. During model training, we utilized the DIV2K training set [41], comprising 800 high-quality 2 K resolution images, to train our CHASE. The model's

performance was assessed using the DIV2K test set (100 2 K images) [41] and a validation set consisting of 120 images randomly selected from the COCO test set [42] and the ImageNet test set [43], respectively. Data augmentation during training involved random horizontal and vertical flipping, as well as random cropping of images. For the test set, images underwent centre cropping exclusively, and the resolution size for both training and test images was maintained at 256×256.

The forward hiding network and the backward reconstructing network underwent training and validation with 16 invertible blocks, respectively, optimizing their parameters using the Adam [44] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In stage 1, the total number of epochs was set to 3 K, with an initial learning rate of 1×10^{-4}, halved every 1 K iterations. Hyperparameters in Equation (11) are configured as $\lambda_h = 8$, $\lambda_r = 1$, $\lambda_l = 1$. In stage 2, the total number of epochs was 1.5 K, the initial learning rate was 1×10^{-4}, and the discriminator underwent pre-training for 500 epochs. The learning rate was halved every 200 epochs, followed by joint training of the discriminator and CHASE for 1 K epoch, with the learning rate halved every 500 epochs. The hyperparameters in Equation (13) were set as $\lambda_h = 16, \lambda_r = 1$, $\lambda_{gan} = 1$. The hiding loss employed an $L2$ regularized loss function, while the reconstruction loss utilized an $L1$ regularized loss function.

Benchmarks. To assess the efficacy of our steganography method, we conducted a comparative analysis against various state-of-the-art image steganography methods, including a version of our proposed steganography method without image permutation algorithm named CHASE_WO, a spatial-domain image steganography method known as 4bit-LSB, Rehman's steganography [22] based on convolutional neural network, and DeepMIH founded on the invertible neural network proposed by Guan [34].

To ensure fairness in the comparison, we acquired the relevant open-source code for each method and adhered to the parameter configurations specified in their respective papers. The retraining process was executed using the aforementioned dataset.

Evaluation metrics. To gauge the quality of the cover-stego image pairs and the secret-reconstructed secret image, we employed two widely used evaluation metrics in image steganography, namely the peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM) [45].

The PSNR serves as a standard image quality metric to assess the impact of a steganography method on an image. It calculates the peak signal-to-noise ratio between the original image and the steganographic image, providing a quantitative measure of image quality. Higher PSNR values indicate superior image quality, with results typically expressed in decibels (dB). The mathematical expressions for PSNR, given an original image X and a generated image Y with an input size of $m \times n$, are as follows:

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [X(i,j) - Y(i,j)]^2 \tag{14}$$

$$\mathrm{PSNR} = 10 \cdot log_{10} \left(\frac{MAX_X^2}{MSE} \right) \tag{15}$$

In the PSNR formula, MAX_X represents the maximum pixel value of the image, typically 255 in the case of an 8-bit image. It is important to note that the PSNR value used for comparison is the average of the PSNR values calculated for the R, G, and B channels of the two images.

On the other hand, SSIM is a metric that considers not only brightness (luminance contrast) differences but also structural similarity. It assesses the structural similarity between the original image and the steganographic image, taking into account luminance and contrast similarities. The SSIM value ranges from -1 to 1, where a value closer to 1

indicates better image quality. For two input images, X and Y, the mathematical expressions for SSIM are as follows:

$$\mathrm{SSIM}(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\delta_{xy} + C_2)}{(\mu_X^2\mu_Y^2 + C_1)\left(\delta_x^2\delta_y^2 + C_2\right)} \tag{16}$$

In the SSIM formula, μ_x and μ_y represent the pixel means of images X and Y, respectively, while δ_x and δ_y are the variances of X and Y. The covariance of X and Y is denoted by δ_{xy}. Additionally, there are two constants, C_1 and C_2, which play a role in preventing division by zero.

Relative Payload (RP) is utilized as an evaluation metric to compare the steganographic capacity of each steganographic image. RP is calculated as the ratio of the amount of secret information that a steganography method can embed to the cover image capacity. This metric gauges the efficiency of information hiding in the image steganography. A higher relative payload signifies that the method can embed more secret information in the image. The mathematical expression for RP is as follows:

$$\mathrm{RP} = \frac{bits(secret\ information)}{bits(cover\ capacity)} \times 100\% \tag{17}$$

Furthermore, we assess the security of our CHASE and benchmarks in Section 4.4 using two prominent types of steganalysis tools: statistical steganalysis and deep learning-based steganalysis.

4.2. Comparison

Quantitative results. Table 3 presents the evaluation metrics for CHASE, as well as the un-scrambling version CHASE_WO, alongside other steganography methods on the DIV2K, COCO, and ImageNet test sets. Notably, CHASE_WO outperformed existing state-of-the-art high-capacity image steganography methods and even low-capacity alternatives for cover-stego image pairs on various datasets at equivalent capacities. While CHASE introduced some degradation in the quality of the stego image, it achieved higher quality for the reconstructed secret image compared to low-capacity steganography methods. Specifically, CHASE_WO improved PSNR by 1.85 dB for cover-stego image pairs, 2.55 dB for secret-reconstructed secret image pairs, and 0.02 in SSIM compared to DeepMIH [34].

In our CHASE, both cover-stego image pairs and secret-reconstructed secret image pairs exhibited PSNR values surpassing 30 dB, indicating excellent visual quality. The PSNR and SSIM of secret-reconstructed secret image pairs were improved by up to 1.43 dB and 0.02, respectively, on the COCO dataset compared to low-capacity steganography methods, showcasing competitive image quality advantages.

Qualitative results. Figure 6 displays images generated by different steganography methods, including 4bit-LSB, Rehman et al. [22], DeepMIH [34], our CHASE, and CHASE_WO. Visually, 4bit-LSB exhibited noticeable artifacts, Rehman et al. produced a yellowish tint, while CHASE, after image scrambling, closely resembled the corresponding cover image. The secret image obtained from the reduction process was nearly identical to the original secret image, showcasing high-fidelity results even with large capacity and complex hidden information.

Table 3. A comprehensive comparison of benchmarks across different datasets. Note for steganography methods with a relative payload greater than 100%, the evaluation metrics for cover-stego images consider the images after hiding, while the metrics for secret-reconstructed secret images represent the average across all extracted secret images. Arrows (↑) indicate the direction of more effective changes in values.

Methods	RP	Cover/Stego Image Pair					
		DIV2K		COCO		ImageNet	
		PSNR (dB) ↑	SSIM ↑	PSNR (dB) ↑	SSIM ↑	PSNR (dB) ↑	SSIM ↑
4bit-LSB	50%	33.19	0.94	33.79	0.94	33.68	0.94
Rehman et al. [22]	33.3%	30.70	0.92	30.18	0.91	32.68	0.93
DeepMIH [34]	300%	34.13	0.94	34.29	0.94	33.39	0.93
CHASE_WO	300%	35.98	0.94	33.59	0.92	33.34	0.93
CHASE	300%	33.09	0.91	31.34	0.90	30.03	0.92
Methods	**RP**	**Secret/Reconstructed Image Pair**					
		DIV2K		COCO		ImageNet	
		PSNR (dB) ↑	SSIM ↑	PSNR (dB) ↑	SSIM ↑	PSNR (dB) ↑	SSIM ↑
4bit-LSB	50%	30.81	0.90	32.04	0.91	31.26	0.90
Rehman et al. [22]	33.3%	32.11	0.93	32.13	0.92	34.75	0.93
DeepMIH [34]	300%	33.47	0.93	33.87	0.93	32.21	0.92
CHASE_WO	300%	36.02	0.95	34.41	0.94	32.07	0.93
CHASE	300%	32.23	0.93	33.47	0.93	31.62	0.91

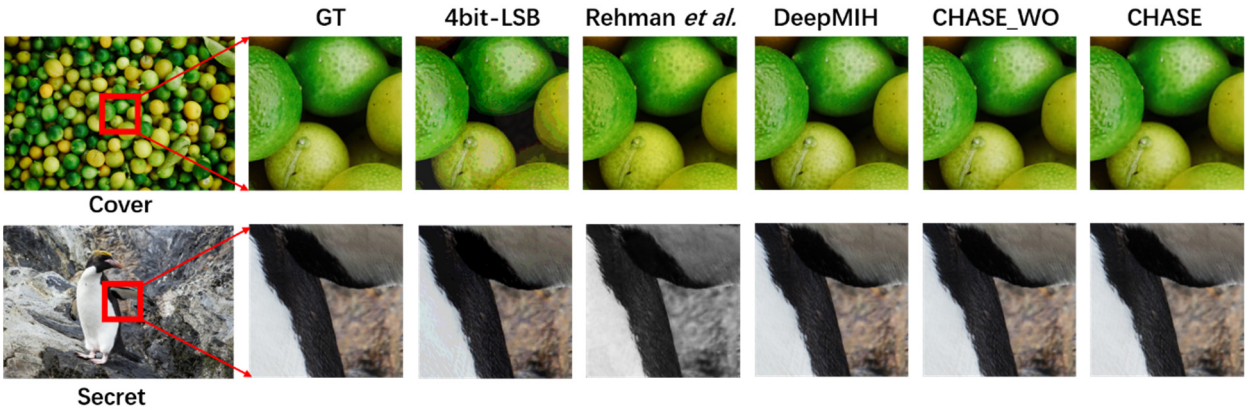

Figure 6. A visual comparison of our CHASE with 4bit-LSB, Rehman et al. [22], and DeepMIH [34], showcasing both the stego images and the reconstructed secret images. The red boxes in the first column denote cropped regions for image steganography actually labeled as GT (Ground Truth) in the second column. The subsequent columns (from the third to seventh) display the steganographic image of different methods. The first row illustrates the effects of their stego images, while the second row shows the reconstructed secret images.

4.3. Ablation Study

Effectiveness of image-permutation. The image disambiguation algorithm serves two crucial purposes: (1) preventing the disclosure of secret information through the corresponding error image, even if the training image set is leaked, and (2) ensuring that, without the key and inverse permutation algorithm knowledge, attackers cannot accurately reconstruct the correct information, even with black-box attack attempts. Table 4 illustrates a noticeable degradation in the image quality of cover-stego and secret-reconstructed image

pairs due to the inclusion of image-permutation. However, this sacrifice is justifiable in scenarios demanding high security.

Table 4. Ablation experiments on image-premutation and GAN.

Image-Premutation	GAN	Cover/Stego Pair	Secret/Reconstructed Pair
×	×	33.82/0.92	35.18/0.93
×	√	35.98/0.94	36.02/0.95
√	×	33.61/0.92	30.41/0.90
√	√	33.09/0.91	32.23/0.93

In Figure 7, the error images produced by various methods highlight the effectiveness of the image-permutation. Despite the challenging task of hiding a secret image with complex texture in a cover image, CHASE's error image exhibited no texture replication artifacts associated with the original secret image. In contrast, error images from 4bit-LSB, Rehman et al. [22], DeepMIH [34], and CHASE_WO showed discernible texture replication artifacts, revealing the outline of the secret information and posing a risk of information leakage. In comparison, CHASE not only achieved a relatively larger hiding capacity but also maintained superior visual effects and security.

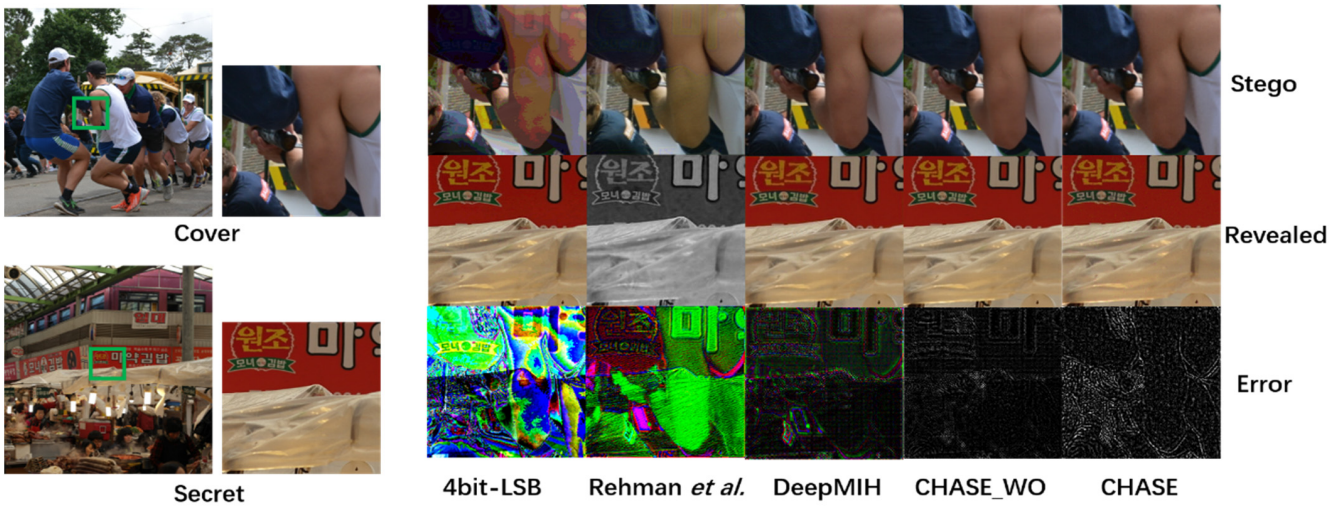

Figure 7. Illustrating a concealed visual comparison on images challenging for steganography. In the cover and secret images in the first column, the green box denotes areas cropped for image steganography. The first row of images on the right displays the stego images for each steganography method [22,34]. The second row on the right showcases the reconstructed secret image, while the third row on the right depicts the error image corresponding to the cover image and the stego image (enhanced 20×).

Effectiveness of the GAN. Table 4 demonstrates a notable enhancement in the image quality of cover-stego and secret-reconstructed image pairs through the incorporation of the GAN. Specifically, in the forward hiding process, the PSNR value for the cover-stego image pair improved by 2.16 dB, and SSIM improved by 0.02. In the backward reconstruction process, the PSNR value for the secret-reconstructed secret image saw an improvement of 0.84 dB, with SSIM increasing by 0.02. The introduction of image-permutation adds complexity to the actual secret image, challenging the model in restoring it. Consequently, the image quality of the secret-reconstructed secret image pair saw a significant decrease. After training with the adversarial training network, the PSNR value and SSIM increased by 1.82 dB and 0.03, respectively. This indicates that GAN effectively promotes the convergence of the distribution of secret images and reconstructed secret images, enhancing image quality while maintaining high restoration accuracy.

4.4. Steganalysis

Steganalysis serves as a countermeasure to steganography, detecting concealed information within carrier media. We evaluated the security of our steganography method against both statistical and deep learning-based steganalysis techniques.

Statistical steganalysis. We employed StegExpose [46], a widely utilized steganalysis tool that relies on statistical image characteristics, to assess the anti-steganalysis capabilities of our CHASE method. To evaluate its performance, we generated 500 stego images using CHASE and selected an additional 1000 clean images for the test set. The StegExpose tool was applied with threshold adjustments between 0 and 1. The true positive rate (TPR) and false positive rate (FPR) were calculated, leading to the creation of the receiver operating characteristic curve (ROC) shown in Figure 8. The orange solid line represents our CHASE method's ROC, while the dark blue dotted line represents random guessing method. The assessment of steganography method detection by steganalysis tools can be framed as a two-classification task, with the Area Under the Receiver Operating Characteristic (ROC) Curve, denoted as AUC, serving as a widely used metric for evaluating the performance of such models. A higher AUC value, closer to 1, indicates superior detection performance, while a value nearing 0.5 suggests performance akin to random guessing. The AUC value of our CHASE method, at 0.53, underscores its strong resistance to steganalysis, affirming its high level of security. StegExpose hardly detect whether secret information was present in the stego images generated by CHASE.

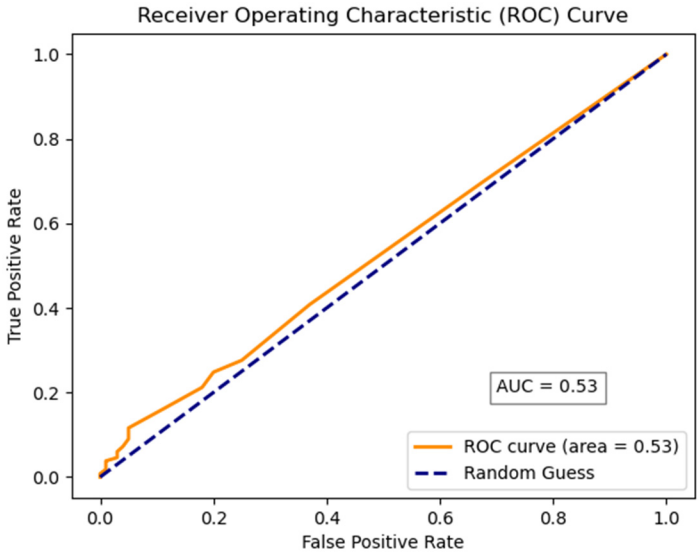

Figure 8. Using the StegExpose tool to generate an ROC curve for our CHASE.

Deep learning-based steganalysis. The rise of deep learning has advanced steganalysis technology. In this section, we employed SRNet [47] and Zhu-Net [48], two state-of-the-art deep learning-based image steganalysis networks, to evaluate our CHASE method and benchmarks. Differing from the conventional approach of training the network with common steganography algorithms and then utilizing the model to detect different methods, we adopted a novel image steganalysis method proposed by Weng et al. [49]. This method involves directly using distinct cover-stego image pairs generated by each steganography method as training sets for the network. The number of training sets was progressively increased to examine the network's detection capabilities under varying quantities of training sets.

We generated 1500 stego images through both CHASE_WO and CHASE, forming their respective test sets with corresponding clean cover images. During the training of SRNet and Zhu-Net, the batch was set to 1, and other hyperparameters were configured according to their respective papers. Figure 9 illustrates changes in the detection accuracy

of the steganalysis model as the number of training sets increases. The left image displays the detection effect under SRNet, while the results for Zhu-Net are on the right.

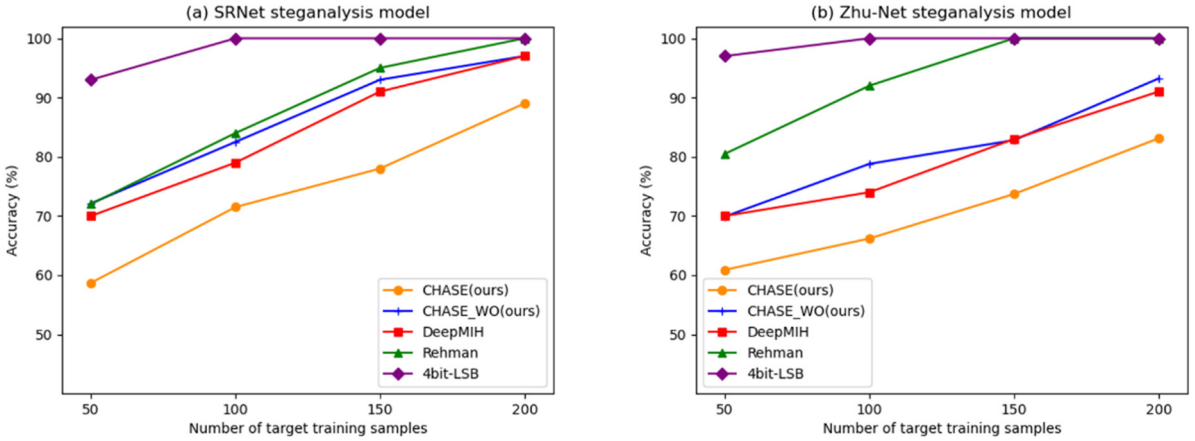

Figure 9. Illustrating the anti-steganalysis capabilities of our CHASE and benchmarks [22,34] by progressively increasing the number of training samples. Panel (**a**) presents the results on the SRNet steganalysis model, while panel (**b**) displays the outcomes on Zhu-Net.

Regardless of SRNet or Zhu-Net, our CHASE method exhibited superior anti-steganalysis capabilities compared to other methods. Under SRNet, even with 200 pairs of training samples, the detection accuracy of our CHASE method surpassed existing methods, with the SOTA method DeepMIH showing an 8% lower accuracy and Zhu-Net showing a 7.9% lower accuracy. Additionally, the unscrambling version CHASE_WO demonstrated an average 11.86% and 10.2% higher detection accuracy on SRNet and Zhu-Net, respectively, compared to the scrambling version CHASE. This discrepancy may be attributed to the distinct deep learning methodologies employed by SRNet and Zhu-Net to analyse residual information in the cover/stego image for prediction. As depicted in Figure 9, compared to other steganography methods, obtaining the residual image by CHASE from the cover/stego image is challenging for an attacker, resulting in enhanced resistance against steganalysis, even at large-capacity steganography scales.

5. Conclusions

In this paper, we present CHASE, an advanced image steganography network based on invertible neural networks, surpassing existing state-of-the-art methods. CHASE achieves a ground-breaking steganography capacity by enabling the concealment of colour images within grayscale images. The incorporation of adversarial networks significantly enhances the imperceptibility of concealed images and restores the quality of hidden information. To bolster security, we introduce an image permutation algorithm based on Logistic chaotic mapping, mitigating the risk of secret information exposure and endowing the steganography network with superior resistance to steganalysis compared to other methods. CHASE exhibits robust generalization across diverse datasets, outperforming other state-of-the-art methods in various numerical comparisons, including capacity and image quality.

Looking ahead, exploring image steganography methods applicable in practical scenarios becomes a crucial direction for research. Real transmission channels often involve malicious eavesdroppers, channel noise, and image compression, posing challenges to secret information transmission. Addressing these challenges and concurrently enhancing capacity, security, and robustness in practical applications represent a pertinent and necessary future research direction.

Author Contributions: Management and supervision, L.H. (Lin Huo) and J.W.; methodology, investigation, and writing—original draft, L.H. (Lin Huo) and R.C.; resources, J.W.; software, R.C.;

validation and visualization, R.C. and L.H. (Lang Huang). All authors have read and agreed to the published version of the manuscript.

Funding: This research presented in this work was supported by Open Project Program of Guangxi Key Laboratory of Digital Infrastructure (Project number: GXDIOP2023007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Subramanian, N.; Elharrouss, O.; Al-Maadeed, S.; Bouridane, A. Image steganography: A review of the recent advances. *IEEE Access* **2021**, *11*, 23409–23423. [CrossRef]
2. Cheddad, A.; Condell, J.; Curran, K.; Mc Kevitt, P. Digital image steganography: Survey and analysis of current methods. *Signal Process.* **2010**, *90*, 727–752. [CrossRef]
3. Evstutin, O.; Melman, A.; Meshcheryakov, A. Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions. *IEEE Access* **2020**, *8*, 166589–166611. [CrossRef]
4. Tamimi, A.A.; Abdalla, A.M.; Al-Allaf, O. Hiding an image inside another image using variable-rate steganography. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **2013**, *4*, 18–21.
5. Asad, M.; Gilani, J.; Khalid, A. An enhanced least significant bit modification technique for audio steganography. In Proceedings of the International Conference on Computer Networks and Information Technology, Abbottabad, Pakistan, 11–13 July 2011; pp. 143–147.
6. Ruanaidh, J.J.K.O.; Dowling, W.J.; Boland, F.M. Phase watermarking of digital images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 19 September 1996; Volume 3, pp. 239–242.
7. Hsu, C.T.; Wu, J.L. Hidden digital watermarks in images. *IEEE Trans. Image Process.* **1999**, *8*, 58–68.
8. Barni, M.; Bartolini, F.; Piva, A. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Trans. Image Process.* **2001**, *10*, 783–791. [CrossRef]
9. Volkhonskiy, D.; Borisenko, B.; Burnaev, E. Generative adversarial netwoks for image steganography. In Proceedings of the Open Review Conference on Learning Representations(ICLR 2016), San Juan, PR, USA, 2–4 May 2016.
10. Tang, W.; Tan, S.; Li, B.; Huang, J. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Process. Lett.* **2017**, *24*, 1547–1551. [CrossRef]
11. Yang, J.; Liu, K.; Kang, X.; Wong, E.K.; Shi, Y.Q. Spatial image steganography based on generative adversarial network. *arXiv* **2018**, arXiv:1804.07939. [CrossRef]
12. Yang, J.; Ruan, D.; Huang, J.; Kang, X.; Shi, Y.Q. An embedding cost learning framework using GAN. *IEEE Trans. Inf. Forensics Secur.* **2018**, *15*, 839–851. [CrossRef]
13. Hayes, J.; Danezis, G. Generating steganographic images via adversarial training. In Proceedings of the Advances in Neural Information Processing Systems, Los Angeles, CA, USA, 4–9 December 2017; pp. 1954–1963.
14. Wang, Y.; Niu, K.; Yang, X. Information hiding scheme based on generative adversarial network. *J. Comput. Appl.* **2018**, *38*, 2923–2928.
15. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2069–2079.
16. Baluja, S. Hiding images within images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1685–1697. [CrossRef]
17. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1–13. [CrossRef]
18. Holub, V.; Fridrich, J. Designing steganographic distortion using directional filters. In Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security, Tenerife, Spain, 2–5 December 2012; pp. 234–239.
19. Pevný, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In Proceedings of the International Workshop on Information Hiding, Calgary, AB, Canada, 28–30 June 2010; pp. 161–177.
20. Shi, H.; Dong, J.; Wang, W.; Qian, Y.; Zhang, X. SSGAN: Secure steganography based on generative adversarial networks. In Proceedings of the Pacific Rim Conference on Multimedia, Harbin, China, 28–29 September 2017; pp. 534–544.
21. Zhang, K.A.; Cuesta-Infante, A.; Xu, L.; Veeramachaneni, K. SteganoGAN:High capacity image steganography with gans. *arXiv* **2019**, arXiv:1901.03892. [CrossRef]
22. Rehman, R.; Nadeem, S.; Nadeem, M.S.; ul Hussain, S. End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography. *arXiv* **2017**, arXiv:1711.07201. [CrossRef]
23. Zhang, R.; Dong, S.; Liu, J. Invisible steganography via generative adversarial networks. *Multimed. Tools Appl.* **2019**, *78*, 8559–8575. [CrossRef]
24. Dinh, L.; Krueger, D.; Bengio, Y. Nice: Nonlinear independent components estimation. *arXiv* **2014**, arXiv:1410.8516. [CrossRef]

25. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density estimation using real NVP. In Proceedings of the 5th International Conference on Learning Representations(ICLR 2017), Toulon, France, 24–26 April 2017.
26. Kingma, D.P.; Dhariwal, P. Glow: Generative flow with invertible 1×1 convolutions. *arXiv* **2018**, arXiv:1807.03039. [CrossRef]
27. Lugmayr, A.; Danelljan, M.; Van Gool, L.; Timofte, R. SRFlow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 715–732.
28. Ardizzone, L.; Lüth, C.; Kruse, J.; Rother, C.; Köthe, U. Guided image generation with conditional invertible neural networks. *arXiv* **2019**, arXiv:1907.02392. [CrossRef]
29. Denker, A.; Schmidt, M.; Leuschner, J.; Maass, P. Conditional Invertible Neural Networks for Medical Imaging. *J. Imaging* **2021**, *7*, 243. [CrossRef] [PubMed]
30. Xing, Y.; Qian, Z.; Chen, Q. Invertible Image Signal Processing 2021. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
31. Xiao, M.; Zheng, S.; Liu, C.; Wang, Y.; He, D.; Ke, G.; Bian, J.; Lin, Z.; Liu, T.Y. Invertible image rescaling. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
32. Lu, S.P.; Wang, R.; Zhong, T. Large-Capacity Image Steganography Based on Invertible Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
33. Jing, J.; Deng, X.; Xu, M.; Wang, J.; Guan, Z. Hinet: Deep image hiding by invertible network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021.
34. Guan, Z.; Jing, J.; Deng, X. DeepMIH: Deep Invertible Network for Multiple Image Hiding. *IEEE Trans. Pattern Anal. Mach. Intell* **2023**, *45*, 372–390. [CrossRef] [PubMed]
35. Gu, G.S.; Liu, F.C. Contourlet domain image encryption based on chaos on mapping. *J. Comput. Appl.* **2011**, *31*, 771–773. [CrossRef]
36. May, R.M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [CrossRef] [PubMed]
37. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 4.
38. Zhang, C.; Benz, P.; Karjauv, A.; Sun, G.; Kweon, I.S. UDH: Universal Deep Hiding for Steganography Watermarking, and Light Fieled Messaging. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 10223–10234.
39. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
40. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521. [CrossRef]
41. Agustsson, E.; Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 126–135.
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; Volume 5, pp. 740–755.
43. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
44. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. [CrossRef]
45. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
46. Boehm, B. Stegexpose—A tool for detecting LSB Steganography. *arXiv* **2014**, arXiv:1410.6656. Available online: https://github.com/b3dk7/StegExpose (accessed on 16 December 2023).
47. Boroumand, M.; Chen, M.; Fridrich, J. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1181–1193. [CrossRef]
48. Zhang, R.; Zhu, F.; Liu, J.; Liu, G. Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1138–1150. [CrossRef]
49. Weng, X.; Li, Y.; Chi, L.; Mu, Y. Highcapacity convolutional video steganography with temporal residual modeling. In Proceedings of the 2019 on International Conference on Multimedia Retrieval, Ottawa, ON, Canada, 10–13 June 2019; pp. 87–95.