

DenseJIN: Dense Depth Image Steganography Model with Joint Invertible and Noninvertible Mechanisms

Delin Duan, Shuyuan Shen, Songsen Yu, Yibo Yuan, Qidong Zhou, Haojie Lv, Huanjie Lin

Abstract—Image steganography discreetly embeds secret information within a carrier, allowing covert communication and enabling the receiver to extract the concealed data when needed. Previous techniques for image steganography had limitations in achieving imperceptibility and security when dealing with images containing intricate textures. In this paper, we introduce DenseJIN, an innovative model for dense depth image steganography. DenseJIN joins invertible and noninvertible mechanisms to achieve effective and secure information hiding. The invertible component of DenseJIN ensures that the stego image maintains high imperceptibility and security, while the noninvertible component enables high-quality recovery of the secret image. In the invertible component, we employ a dense connection for each invertible block in the forward process and a straightforward series connection during the reverse process. In the forward process of the network, the secret image is embedded, while the backward process is responsible for extracting the embedded secret image. To perform the noninvertible step, we incorporate a modified Unet architecture, enabling deep fine-grained feature extraction from cover images and secret images. Our experimental results indicate that DenseJIN surpasses other contemporary image steganography methods. On average, DenseJIN achieves a remarkable improvement of over 1.75 dB in PSNR for secret image recovery across DIV2K, COCO and ImageNet.

Index Terms—Information security, invertible neural network, image hiding, wavelet transform

1 INTRODUCTION

STEGANOGRAPHY [1] refers to the practice of concealing confidential information within various types of media, such as text, audio, images, and video, without drawing attention to the fact that such information is being hidden. For the image hiding task [2], both the secret information and carrier are images. The image to be hidden is known as the secret image, while the image chosen to host the secret image is referred to as the cover image. The resulting image after applying steganography is called the stego image [3]. The primary objective of image hiding is to employ a dedicated hiding algorithm to conceal one or more secret images within the cover image securely. Consequently, the concealed secret images can be effectively extracted and recovered by the recipient using a specified algorithm. In the hiding process, it is usually desirable for the generated stego [4] image to closely resemble the cover image in terms of visual appearance, ensuring that it remains visually indistinguishable. Furthermore, it is crucial to ensure that the extracted secret image closely resembles the original secret image in terms of visual characteristics.

The image hiding task requires not only high capacity

but also security and invisibility. Although some recent work [5] has improved the visual quality of stego images, their payload capacity is still small, and they cannot balance high steganographic capacity and imperceptibility. Spatial domain-based [6] methods and frequency domain-based [7], [8], [9], [10] methods are the two primary categories into which traditional image steganography methods can be classified. In spatial domain-based methods, the process of information embedding involves altering the pixel values of the carrier image, while frequency domain-based methods modify the frequency coefficients of the transformed cover image. These traditional steganographic methods have a limited capacity of 0.2 to 4 bits per pixel (bpp), while recent learning-based methods [11], [12] strive to increase steganographic capacity and have shown significant improvements. These learning-based methods design separate hiding and recovery networks for embedding and extracting secret information.

Recently, some scholars have tried to use invertible neural networks (INNs) for various reversible problems, such as image compression [13], colorization [14] and rescaling [15]. The hiding and recovery processes in the image hiding task can also be regarded as a pair of reversible problems [3], so the INN is naturally applicable to the image hiding task. By introducing an INN, the invertible steganography network (ISN) [3] achieved a significant increase in hiding capacity. The steganographic capacity can be further increased by increasing the number of channels in the hidden network branch. HiNet [2] employed frequency domain transformation to process images, allowing the effective concealment of secret information. To

- This work is supported by the Guangdong Basic and Applied Basic Research Fund (No. 2023A1515011472.) (Corresponding author: Shuyuan Shen.)
- Delin Duan, Shuyuan Shen, Songsen Yu, Yibo Yuan, Qidong Zhou, Haojie Lv and Huanjie Lin are with the School of Software, South China Normal University, FoShan 528225, China (e-mail: delinduan@m.scnu.edu.cn; ssyuan16@m.scnu.edu.cn; yss8109@163.com; yiboyuan@m.scnu.edu.cn; zhouqidong@m.scnu.edu.cn; hjloo97@m.scnu.edu.cn; linhuanjie@m.scnu.edu.cn).

Manuscript received April 19, 2005; revised August 26, 2015.

improve the safety of steganography, HiNet [2] introduced a meticulously designed low-frequency wavelet loss function that enables precise control over the distribution of secret information among various frequency components.

The aforementioned INN-based steganographic scheme provides a new steganographic design idea that not only expands steganographic capacity but also improves security. While there has been progress in achieving invisibility and enhancing security for stego images, there are still limitations in regard to complex textures. First, the invertible blocks in INNs are simply connected to each other in series, and this loose connection may lead to information loss. This further makes the generated stego image vulnerable to texture replication artifacts. Second, the ability of the generated stego images to resist steganalysis attacks is also relatively poor. Finally, it is important to acknowledge that the omission of certain deep feature information from cover images and secret images can adversely affect the quality of the recovered secret image.

Addressing these challenges, this paper introduces DenseJIN, a novel dense depth steganography framework that contains jointly invertible and noninvertible mechanisms. This paper makes the following key contributions:

- To enhance the imperceptibility of the stego image, we densely connect the individual invertible blocks in the model. Utilizing dense connections, our approach aims to minimize information loss and generates a stego image closely resembling the original cover image in appearance.
- To improve the quality of recovered secret image, we incorporate a noninvertible network (modified Unet) into the hiding process. By focusing on extracting deep features from secret images, this network significantly contributes to the recovery process, ultimately resulting in the recovery of a secret image of superior quality.
- To comprehensively evaluate our approach, we conducted a series of experiments on diverse image datasets, comparing the performance of our approach with that of existing sota image hiding approaches. Our experimental outcomes unequivocally indicate that our approach surpasses existing approaches in terms of both imperceptibility and security. These findings highlight the efficacy and reliability of our approach for image hiding applications.

The structure of the paper unfolds as follows: Section 2 offers a comprehensive review of existing research on image hiding and INNs. Section 3 delineates the complete network architecture of DenseJIN and outlines our adopted training strategies. Section 4 delves into the experimental findings, while Section 5 encapsulates our conclusions drawn from these results.

2 RELATED WORKS

2.1 Image Hiding.

Unlike data encryption which focuses on securing information through encryption algorithms, steganography takes a different approach by embedding secret data within a host, aiming to conceal the presence of such data. Image hiding is a significant research area within the field of steganography and is primarily concerned with concealing an entire image within another image [2]. Image steganography aims to embed a confidential image within a cover image using an algorithm, producing a stego image. Subsequently, a recovery algorithm is applied to extract the secret image from the stego image. For the best outcomes, it's vital that the stego image closely mirrors the cover image, demonstrating a significant degree of likeness. Moreover, the stego image should possess the capability to effectively evade detection by steganalysis techniques [16], [17]. The recovered secret image should also strongly resemble the original secret image, ensuring accurate and faithful recovery.

Image steganography encompasses both spatial and frequency domain methods, with LSB [6] representing a classic technique in the spatial domain. The LSB method has certain limitations, such as a low payload capacity and a tendency to produce texture replication artifacts in smooth areas of the cover image. Spatial domain-based steganography approaches include histogram shifting [18], pixel value differencing (PVD) [19] and palettes [20], [21]. Frequency domain-based steganography methods are commonly employed to conceal images by performing transformations in the frequency domain using techniques such as the discrete Fourier transform (DFT) [22], discrete wavelet transform (DWT) [23], and discrete cosine transform (DCT) [24]. Since the phase is more important for image comprehensibility than the amplitude of DFT values, Ruanaidh et al. [22] proposed utilizing the phase of DFT to embed information. Zhao et al. [23] introduced a method that utilizes the chaotic wavelet domain to embed secret information into the DWT coefficients of a subimage. Tsu et al. [24] suggested concealing the secret information in the mid-frequency subband coefficients. Frequency domain-based methods typically have low steganographic capacity but demonstrate strong resistance against interference operations.

The utilization of deep learning techniques for image steganography has garnered significant attention from researchers in recent years. Baluja [25] first proposed hiding color images within carrier images of the same dimensions with a neural network model. In an attempt to enhance the hiding capacity, he explored the concealment of 2 secret images in succession within a single cover image and introduced a multi-images concealing framework. Duan et al. [12] suggested a Unet-based image steganography scheme. Based on this work, Liu et al. [11] presented a new framework combining discrete wavelet transform for segmenting image details with enhanced steganography. Some GAN-based methods have also been proposed. The hidden network put forward by Zhu et al. [5] and

SteganoGAN presented by Zhan et al. [26] added a network to act as an adversary in addition to training the encoder network and decoder network for information embedding and extraction to enhance the resistance to steganalysis.

2.2 Normalizing Flow-based Model.

The INN is a powerful generative model presented by Dinh et al. [27], which aims to learn a bijective mapping between the source domain x and the target domain z . Specifically, it uses the bijective function F_θ to map a complex data distribution $p_X(x)$ to a simple prior distribution $p_Z(z)$; that is, $z = F_\theta(x)$. Then, we can generate x as $x = F_\theta^{-1}(z)$, where θ is a parameter shared by the forward and backward processes. Thus we can define the model distribution on X by:

$$p_X(x) = p_Z(F_\theta(x)) \left| \det \left(\frac{\partial F_\theta(x)}{\partial x^T} \right) \right|, \quad (1)$$

where $\left| \det \left(\frac{\partial F_\theta(x)}{\partial x^T} \right) \right|$ is the absolute value of the determinant of the Jacobian matrix $\frac{\partial F_\theta(x)}{\partial x^T}$. To reduce the computational cost of computing the Jacobian matrix determinant, Dinh et al. introduced an additive layer and an affine coupling layer into NICE [27] and RealNVP [28], respectively. Then Kingma et al. [29] proposed the Glow which introduces a more flexible reversible 1×1 convolutional layer to replace the fixed permutation layer in RealNVP.

Owing to its excellent fitting capabilities, the INN has gained substantial recognition in the realm of image processing and has found applications in numerous challenging tasks. Specifically, Lugmayr et al. [30] presented SRFlow which is a method based on the normalizing flow of image superresolution. Xie et al. [31] presented an enhanced invertible encoding model with an INN, which effectively mitigates the issue of information loss and achieves superior compression performance. There are also several other tasks, such as image scaling [15], image denoising [32], image coloring [14] and unexposed image enhancement [33], which INNs can perform. Considering that concealing and retrieving secret images in image hiding scenarios involve invertible processes, the utilization of INNs in this context is intuitive. The ISN proposed by Lu et al. [3] simply utilized several tandem invertible blocks to achieve effective hiding and recovery. Building upon this foundation, HiNet [2], proposed by Jing et al., leveraged wavelet domain transformation for image hiding and introduced a low-frequency wavelet loss function. This innovative approach aimed to enhance security by concealing secret information in the high-frequency region. Subsequently, to achieve higher steganographic capacity, Jing et al. proposed DeepMIH [34], a multi-image steganographic model that incorporated an importance map module to dynamically direct the embedding process of each image according to the outcomes of the prior image hiding step. Cheng et al. [35] introduced a comprehensive framework called IICNet for reversible transformation. This framework incorporated a relation module to enhance

the nonlinearity of the INN, facilitating the extraction of cross-image relations. Moreover, a channel squeeze layer was integrated to amplify the network's flexibility. Xu et al. [36] introduced RIIS, a robust steganographic framework that incorporates conditional normalizing flow. This innovative approach allows the distribution of redundant high-frequency components to be modeled, taking into account the condition of the cover image. Li et al. [37] introduced iSCMIS which introduced spatial attention and channel attention mechanisms, combining them with the dense block as the basic module of the INNs, respectively, which effectively enhanced the invisibility and security. Mou et al. [38] introduced LF-VSN for video concealment using INNs. Their method not only concealed seven secret videos in a cover video, but also enabled the receiver to recover a specific secret video from the cover video using a specific key. Hu et al. [39] proposed DIH-OAIN. In contrast to HiNet and DeepMIH, this model incorporates a probabilistic guided module, enabling the concealment of the secret image within complex texture regions. Additionally, to enhance the stego image's resistance against steganalysis, they introduced a one-way adversarial training strategy. Experimental results demonstrate that their method enhances both imperceptibility and security while preserving low computational complexity.

3 METHODS

TABLE I: The explanations of the important symbols appearing in this paper. Note that capital X, R and Z denote subbands obtained after DWT processing.

| Symbols | Description |
|---|--|
| $x_{\text{sec}}/X_{\text{sec}}$ | The secret image |
| $x_{\text{cov}}/X_{\text{cov}}$ | The cover image |
| $x_{\text{ste}}/X_{\text{ste}}$ | The stego image |
| $x_{\text{sec_rec}}/X_{\text{sec_rec}}$ | The recovered secret image |
| $x_{\text{cov_rec}}/X_{\text{cov_rec}}$ | The recovered cover image |
| r/R | The information lost in the hiding procedure |
| z/Z | The auxiliary variables |
| $\mathcal{H}(\cdot)/T$ | The haar DWT |
| T^{-1} | The inverse haar DWT |

3.1 Network framework

This section presents the design of DenseJIN, an end-to-end dense network structure, achieved by integrating dense connections and a noninvertible Unet architecture. Fig. 1 illustrates the model framework of DenseJIN, offering a comprehensive overview of its structure, which comprises the following components: a DWT/IDWT module, a dense invertible module (DIM), a noninvertible Unet-based module (NIUM) and a channel squeeze module (CSM). Table I provides explanations for the important symbols utilized in this paper.

The embedding process, known as the forward process, begins with the cover image and the secret image undergoing DWT to obtain their corresponding low-frequency and high-frequency subbands, respectively. Subsequently, these subbands are concatenated and

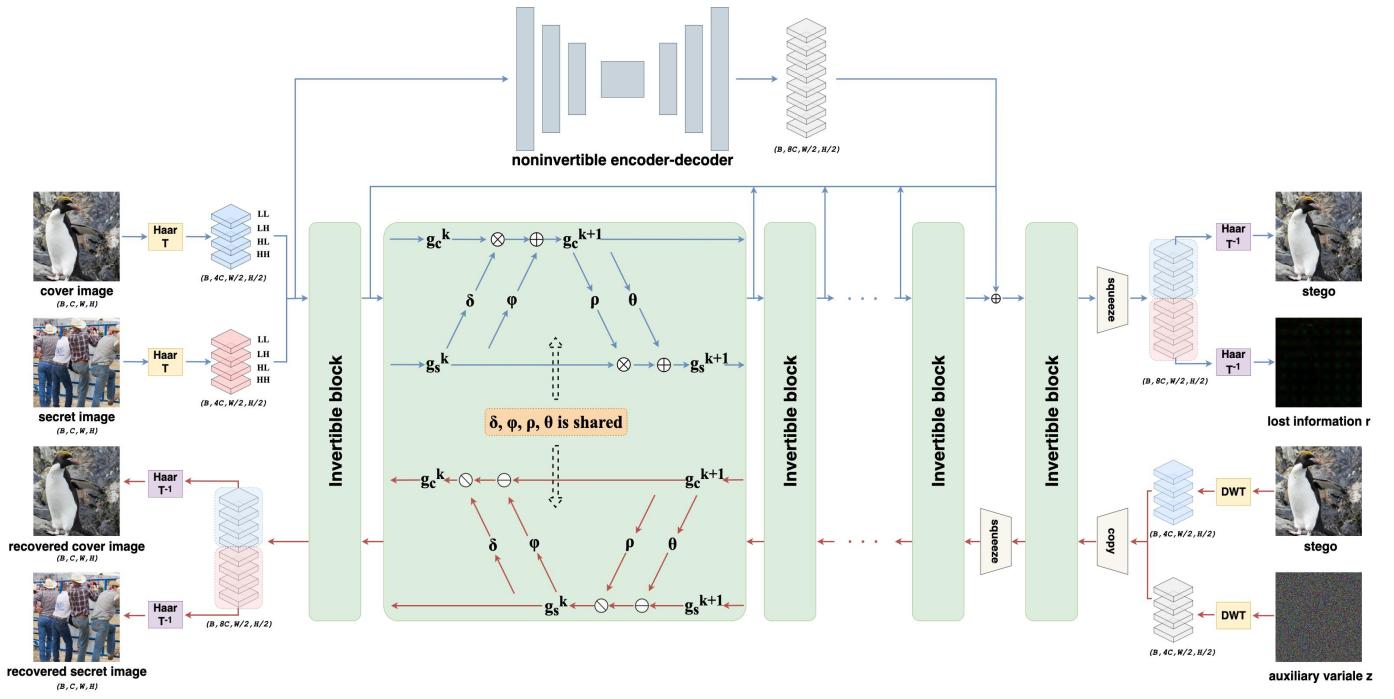


Fig. 1: The framework of DenseJIN, which consists of invertible and noninvertible blocks, where the noninvertible encoding-decoding network is a modified Unet.

inputted into the NIUM and DIM modules, respectively. Notably, the outputs of the NIUM serve as the inputs for the last invertible block in the DIM. The resulting outputs of the DIM are then processed by the CSM to adjust the channel count. Finally, the portion of the outputs from the channel compression layer corresponding to the cover image branch undergoes IDWT to generate the stego image. The formula for the forward embedding process is presented below:

$$(\mathbf{x}_{\text{ste}}, \mathbf{r}) = f_{\epsilon}(\mathbf{x}_{\text{cov}}, \mathbf{x}_{\text{sec}}), \quad (2)$$

where $f_{\epsilon}(\cdot)$ denotes the network function at the forward embedding process, \mathbf{r} denotes the discarded information.

The backward process is the process of secret image extraction and uses the identical network structure and parameters as the forward process. Specifically, the auxiliary variable \mathbf{z} is concatenated to the stego image and then a channel copy operation is performed before passing \mathbf{z} and stego image into the last invertible block of the DIM. Then, the outputs of the final invertible block are compressed and sequentially fed into the remaining invertible blocks to perform affine coupling transformation. Finally, the recovered cover and secret images are obtained by taking the tensor of the cover and secret branches of the outputs from the DIM and applying the IDWT, respectively. It's crucial to emphasize that the NIUM is only involved at the embedding process and not in the extraction process.

$$(\mathbf{x}_{\text{cov_rev}}, \mathbf{x}_{\text{sec_rev}}) = f_{\epsilon}^{-1}(\mathbf{x}_{\text{ste}}, \mathbf{z}), \quad (3)$$

where $f_{\epsilon}^{-1}(\cdot)$ represents the network function at the backward extraction process, \mathbf{z} is obtained by random sampling

from a Gaussian distribution to substitute the discarded information \mathbf{r} at the embedding process. We provide detailed steps for the embedding and extraction process in Algorithms 1, 2 to further explain the working process of our proposed DenseJIN.

Algorithm 1: The embedding process.

Input: The cover image \mathbf{x}_{cov} , the secret image \mathbf{x}_{sec} and model I_{Φ} .

Output: The stego image \mathbf{x}_{ste} .

- 1 Compute $(\mathbf{X}_{\text{cov}}, \mathbf{X}_{\text{sec}}) = \mathbf{T}(\mathbf{x}_{\text{cov}}, \mathbf{x}_{\text{sec}})$;
 - 2 Compute $(\mathbf{X}_{\text{ste}}, \mathbf{R}) = I_{\Phi}(\mathbf{X}_{\text{cov}}, \mathbf{X}_{\text{sec}})$;
 - 3 Compute $(\mathbf{x}_{\text{ste}}, \mathbf{r}) = \mathbf{T}^{-1}(\mathbf{X}_{\text{ste}}, \mathbf{R})$;
 - 4 **return** \mathbf{x}_{ste}
-

Algorithm 2: The extraction process.

Input: The stego image \mathbf{x}_{ste} and model I_{Φ} .

Output: The secret image $\mathbf{x}_{\text{sec_rev}}$.

- 1 Compute $\mathbf{z} = \text{Gauss}(\mathbf{x}_{\text{ste}})$;
 - 2 Compute $(\mathbf{X}_{\text{ste}}, \mathbf{Z}) = \mathbf{T}(\mathbf{x}_{\text{ste}}, \mathbf{z})$;
 - 3 Compute $(\mathbf{X}_{\text{cov_rev}}, \mathbf{X}_{\text{sec_rev}}) = I_{\Phi}^{-1}(\mathbf{X}_{\text{ste}}, \mathbf{Z})$;
 - 4 Compute $(\mathbf{x}_{\text{cov_rev}}, \mathbf{x}_{\text{sec_rev}}) = \mathbf{T}^{-1}(\mathbf{X}_{\text{cov_rev}}, \mathbf{X}_{\text{sec_rev}})$;
 - 5 **return** $\mathbf{x}_{\text{sec_rev}}$
-

3.2 DWT/IDWT Module

Similar to HiNet [2] and DeepMIH [34], our approach leverages the frequency domain to hide secret information. This choice is motivated by the fact that embedding

information in the pixel domain can often result in undesirable texture replication artifacts and chromatic distortion [40], [41]. As shown in Fig. 1, the cover image and secret image are split into low-frequency and high-frequency wavelet subbands, respectively. Since the wavelet transform has perfect reconstruction properties and can help mitigate information loss, we employ DWT/IDWT to convert the image to the frequency domain for information embedding.

After applying the DWT, the input images are transformed into wavelet subbands. These subbands have dimensions of half the height and width of the original image, but their channel number is four times that of the original image. Due to the bidirectional symmetry of the wavelet transform, the wavelet subbands can be perfectly reconstructed back into the original image through the IDWT. By skillfully integrating the DWT and IDWT into our method with careful design considerations, we ensure the smooth coexistence of these operations with our model's end-to-end training without any adverse effects. The DWT/IDWT operations in the forward and backward processes can be expressed as follows:

$$(\mathbf{X}_{\text{cov}}, \mathbf{X}_{\text{sec}}) = \mathbf{T}(\mathbf{x}_{\text{cov}}, \mathbf{x}_{\text{sec}}), \quad (4)$$

$$(\mathbf{x}_{\text{ste}}, \mathbf{r}) = \mathbf{T}^{-1}(\mathbf{x}_{\text{ste}}, \mathbf{R}), \quad (5)$$

$$(\mathbf{X}_{\text{ste}}, \mathbf{Z}) = \mathbf{T}(\mathbf{x}_{\text{ste}}, \mathbf{z}), \quad (6)$$

$$(\mathbf{x}_{\text{cov_rev}}, \mathbf{x}_{\text{sec_rev}}) = \mathbf{T}^{-1}(\mathbf{x}_{\text{cov_rev}}, \mathbf{X}_{\text{sec_rev}}), \quad (7)$$

3.3 Dense Invertible Module

The reversibility of INNs makes them naturally suited for image steganography tasks. The shared parameters in the invertible block of the embedding and extraction process contribute to the production of high-quality images in methods based on INNs. In contrast to the current INN-based methods [2], [3], [34], [35], [37] which connect each invertible block in series, we employ dense connections to link each invertible block. To be more precise, in our design, we utilize a dense connection during the forward embedding process and opt for a sequential connection during the backward extraction process. Fig.1 depicts the formation of DIM, taking inspiration from the DeepMIH [34] framework as a basis. Specifically, assuming there are n invertible blocks, the input of the first $n - 1$ invertible blocks in the forward process is the output of the previous invertible block, which can be expressed as follows:

$$\mathbf{X}_{\text{IB}}^k = IB_k(\mathbf{X}_{\text{IB}}^{k-1}), \quad (8)$$

where $IB_k(\cdot)$ denotes the forward process of the k th invertible block, and $\mathbf{X}_{\text{IB}}^{k-1}$ and \mathbf{X}_{IB}^k denote the outputs of the $(k - 1)$ th and k th invertible blocks, respectively. For the n th invertible block, its input is the concatenation of the output of the previous $n - 1$ invertible blocks with the

output of NIUM. The processing of the n th invertible block is shown below:

$$\mathbf{X}_{\text{IB}}^n = IB_n([\mathbf{X}_{\text{IB}}^1, \mathbf{X}_{\text{IB}}^2, \dots, \mathbf{X}_{\text{IB}}^{n-1}, \mathbf{X}_{\text{NIUM}}]), \quad (9)$$

where \mathbf{X}_{NIUM} denotes the output of NIUM, n is the number of invertible blocks used in the DIM. However, in the backward process, we only concatenate the individual invertible blocks and the NIUM is not involved in the computation, the process is described as follows:

$$\mathbf{X}_{\text{IB}}^{k-1} = IB_k^{-1}(\mathbf{X}_{\text{IB}}^k), \quad (10)$$

where IB_k^{-1} denotes the backward process of the k th invertible block.

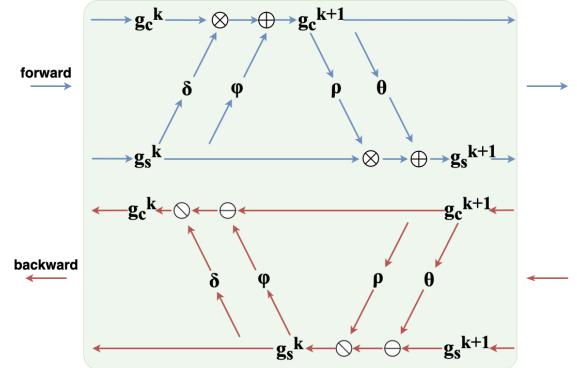


Fig. 2: The structure diagram of the invertible block.

Invertible block. The invertible block in the DIM is shown in Fig.2, in the k -th invertible block, g_c^k and g_s^k denote the cover image branch and secret image branch, respectively. The terms g_c^{k+1} and g_s^{k+1} refer to the outputs of the cover image branch and secret image branch, respectively, following the current affine coupling transformation. The affine coupling transformation can be expressed as follows:

$$g_c^{k+1} = g_c^k \odot \exp(\alpha(\delta(g_s^k))) + \varphi(g_s^k), \quad (11)$$

$$g_s^{k+1} = g_s^k \odot \exp(\alpha(\rho(g_c^{k+1}))) + \theta(g_c^{k+1}), \quad (12)$$

where α is a clamp factor obtained by multiplying by a sigmoid function, and the \odot symbol denotes the Hadamard product operation. The $\exp(\cdot)$ is an exponential operator, and $\delta(\cdot)$, $\varphi(\cdot)$, $\rho(\cdot)$ and $\theta(\cdot)$ are arbitrary functions. Here, we adopt dense blocks which are shown in Fig.3 to represent them. The backward propagation for the extraction process is described as follows:

$$g_s^k = (g_s^{k+1} - \theta(g_c^{k+1})) \odot \exp(-\alpha(\rho(g_c^{k+1}))), \quad (13)$$

$$g_c^k = (g_c^{k+1} - \varphi(g_s^k)) \odot \exp(-\alpha(\delta(g_s^k))), \quad (14)$$

The dense block. As shown in Fig. 3, a dense block consists of multiple densely connected layers used to extract feature information from an image. In each dense block, the outputs of all previous layers are connected to the input of the current layer and processed by a nonlinear transform. Assuming that the output of layer i is x_i , the formula for layer i is:

$$x_i = H_i([x_0, x_1, \dots, x_{(i-1)})], \quad (15)$$

where $H_i(\cdot)$ denotes the nonlinear transformation of the i th layer, which we process using the LeakyReLU activation function. $[x_0, x_1, \dots, x_{(i-1)}]$ denotes the concatenation of the outputs of the first i layers. Table II presents the detailed configuration information of the dense blocks we used.

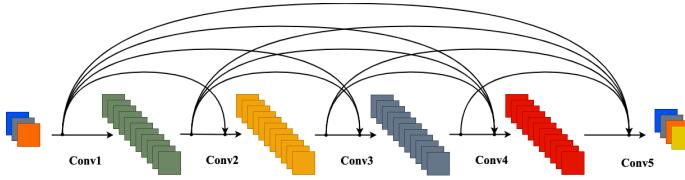


Fig. 3: The structure diagram of a dense block.

TABLE II: Configuration details for the dense block.

| Layer | Input size | Kernel size | Output size |
|-------|--|------------------------|--------------------------|
| input | $12 \times 64 \times 64$ | / | $12 \times 64 \times 64$ |
| Conv1 | $12 \times 64 \times 64$ | $32 \times 3 \times 3$ | $32 \times 64 \times 64$ |
| Conv2 | $(12 + 32) \times 64 \times 64$ | $32 \times 3 \times 3$ | $32 \times 64 \times 64$ |
| Conv3 | $(12 + 32 \times 2) \times 64 \times 64$ | $32 \times 3 \times 3$ | $32 \times 64 \times 64$ |
| Conv4 | $(12 + 32 \times 3) \times 64 \times 64$ | $32 \times 3 \times 3$ | $32 \times 64 \times 64$ |
| Conv5 | $(12 + 32 \times 4) \times 64 \times 64$ | $12 \times 3 \times 3$ | $12 \times 64 \times 64$ |

3.4 Noninvertible Unet-Based Module

In the INNs, the information that is used in the affine coupling operation includes only shallow extracted features, meaning that the obtained INNs can model only the shallow features and thus ignores the information of the complex part of the texture. As depicted in Fig.5, the ISN [3] and HiNet [2] are both designed by using the INNs exclusively. However, the resultant reconstructed secret images suffer from severe texture replication artifacts. Fig.1 shows that our framework utilizes an additional Unet-based noninvertible module which significantly boosts the representation capability of INNs. The structure of the modified Unet is depicted in Fig.4, which consists of an encoder part and a decoder part. The encoder part consists of 4 downsampling modules. And each downsampling module consists of two 3×3 convolutions and a 2×2 max pooling operation. The decoder part consists of 4 upsampling modules, each upsampling module includes a 2×2 transposed convolution, a feature-concatenated operation and two 3×3 convolutions. The encoder component is utilized to extract essential features from the input dataset, whereas the decoder component is tasked with reconstructing these features into the spatial structure of the original input. The working process of NIUM can be formulated as follows:

$$\mathbf{X}_{\text{NIUM}} = \text{NIUM}(\mathbf{X}_{\text{cov}}, \mathbf{X}_{\text{sec}}), \quad (16)$$

where $\text{NIUM}(\cdot)$ denotes the NIUM structure we propose. The NIUM is solely involved in the forward concealing process, extracting deep and detailed features that contribute to a more fine-grained representation. More precisely, as the input and output sizes of the INNs remain fixed, it is imperative that the input and output sizes of the noninvertible Unet module align with them. To achieve this, we made

adjustments to the Unet network structure; in particular, we set the padding to 1 during the 3×3 convolution.

3.5 Loss Functions

The loss functions utilized in our model are inspired by those presented in [2], which include embedding loss, extracting loss and low-frequency wavelet loss.

Embedding loss. To ensure that the stego image is not visually detected as containing secret information, the stego image is required to be as identical as possible to the cover image. The embedding loss L_{emb} is specified as follows,

$$L_{\text{emb}}(\varepsilon) = \sum_{n=1}^N \ell_\omega \left(\mathbf{x}_{\text{cov}}^{(n)}, \mathbf{x}_{\text{ste}}^{(n)} \right), \quad (17)$$

where ε are the parameters of the network, and the stego image $\mathbf{x}_{\text{ste}}^{(n)}$ is generated by $f_\varepsilon \left(\mathbf{x}_{\text{cov}}^{(n)}, \mathbf{x}_{\text{sec}}^{(n)} \right)$, with $f_\varepsilon(\cdot)$ indicating the forward embedding process. The N denotes the number of training samples, and ℓ_ω indicates the ℓ_1 or ℓ_2 norm.

Extracting loss. The recovered secret image which extracted from the stego image must be of visually high quality, and is preferably indistinguishable from the original secret image. The extracting loss L_{ext} is described as follows,

$$L_{\text{ext}}(\varepsilon) = \sum_{n=1}^N E_{z \sim p(z)} \left[\ell_\varsigma \left(\mathbf{x}_{\text{sec}}^{(n)}, \mathbf{x}_{\text{sec_rec}}^{(n)} \right) \right], \quad (18)$$

where the z is sampled from the Gaussian distribution $p(z)$. The recovery image $\mathbf{x}_{\text{sec_rec}}^{(n)}$ is equal to $f_\varepsilon^{-1} \left(\mathbf{x}_{\text{ste}}^{(n)}, z \right)$ where $f_\varepsilon^{-1}(\cdot)$ denotes the backward extraction process. ℓ_ς is ℓ_1 or ℓ_2 norm.

Low-frequency wavelet loss. Similar to [2], we incorporate an extra low-frequency wavelet loss L_{freq} into the framework to enhance the network's resistance to steganalysis. To achieve optimal information concealment in the high-frequency subbands, it is vital to maintain a high degree of similarity between the low-frequency subbands of the stego image and the corresponding subbands of the cover image. This can be achieved by utilizing discrete wavelet transform to obtain the subbands. The low-frequency wavelet loss L_{freq} is described as follows,

$$L_{\text{freq}}(\varepsilon) = \sum_{n=1}^N \ell_\varrho \left(\mathcal{H} \left(\mathbf{x}_{\text{cov}}^{(n)} \right)_{LL}, \mathcal{H} \left(\mathbf{x}_{\text{ste}}^{(n)} \right)_{LL} \right), \quad (19)$$

where $\mathcal{H}(\cdot)_{LL}$ denotes the function that extracts the low-frequency subbands following the Haar DWT. After subjecting both cover and stego images into the Haar DWT, ℓ_ϱ measures the distance between each pair of low-frequency subbands.

Total loss function. In conclusion, optimizing our model involves minimizing the total loss function, denoted as L_{total} which can be expressed as follows,

$$L_{\text{total}} = \lambda_{\text{emb}} L_{\text{emb}} + \lambda_{\text{ext}} L_{\text{ext}} + \lambda_{\text{freq}} L_{\text{freq}}. \quad (20)$$

where λ_{emb} , λ_{ext} and λ_{freq} are the weight coefficients of the embedding loss L_{emb} , extracting loss L_{ext} and low-frequency wavelet loss L_{freq} , respectively. Algorithm 3 provides an overview of the training procedure for our DenseJIN model. All network parameters for DIM and NIUM are updated during training of the whole network.

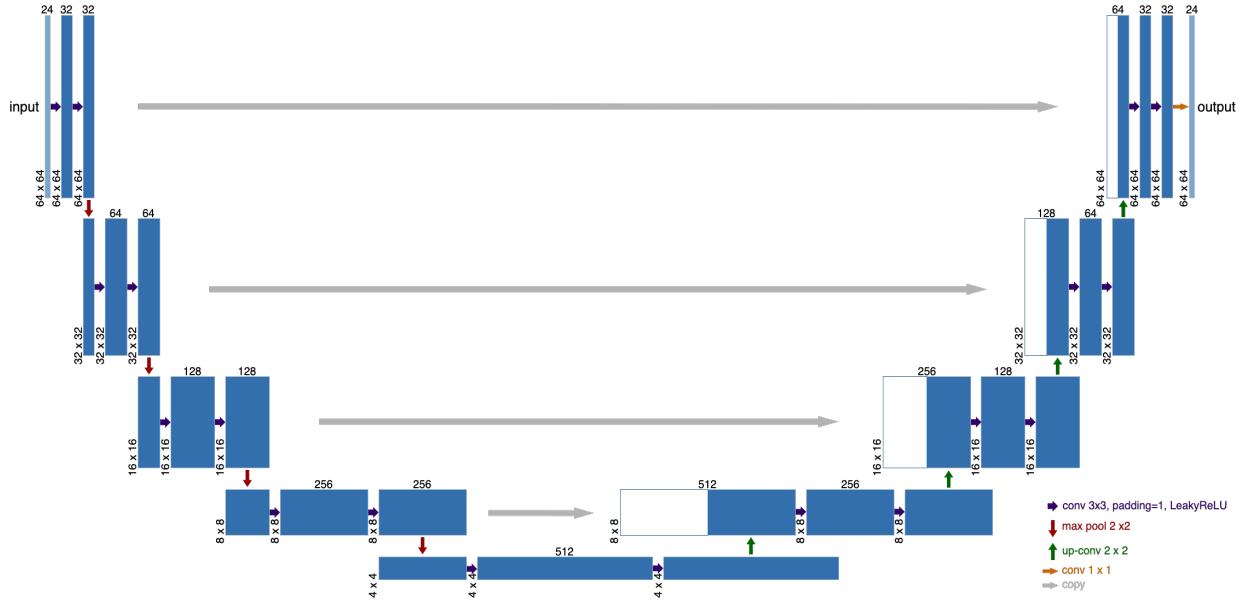


Fig. 4: The structure of the noninvertible encode-decode network (modified Unet) used in our model.

4 EXPERIMENTS

This section empirically evaluates the effectiveness of DenseJIN through a series of experiments. We first outline the experimental settings in Section 4.1. Next, we compare our method's performance with that of other leading techniques in single-image hiding and multi-image hiding in Sections 4.2 and 4.3, respectively. Subsequently, we describe the visualization of DenseJIN in Section 4.4. Following that, we assess DenseJIN's resistance to steganalysis in Section 4.5, comparing it with other methods. In Section 4.6, we conduct ablation experiments to investigate the contribution of individual components in our approach. Finally, performance evaluation is conducted in Section 4.7.

4.1 Experimental Setting

Datasets. Our network is trained using the DIV2K [42] training dataset comprising 800 images, each with a resolution of 128×128 . For testing purposes, we utilize the DIV2K [42] testing dataset comprising 100 images with a resolution of 1024×1024 , along with COCO [43], consisting of 5,000 randomly selected images with a resolution of 256×256 , and ImageNet [44] which includes 50,000 randomly chosen images, also with a resolution of 256×256 . During training, we

use a randomized cropping strategy to prepare the training images. It is important to highlight that we employ a central cropping strategy during the testing phase. This strategy ensures consistent positioning and resolution maintenance for both cover and secret images.

Implementation details. The DenseJIN is developed in PyTorch and accelerated through the use of an Nvidia A30 GPU. To guarantee a fair comparison, we replicated ISN [3], HiNet [2] and IICNet [35] on an Nvidia A30 GPU with the same settings. Our training patches have a size of 128×128 , and we iterate for a total of 500K times. The mini-batch size is set as 8, with an equal distribution of randomly selected cover patches and secret patches, each comprising half of the samples. We use 8 embedding and extracting blocks. The parameters λ_{emb} , λ_{ext} and λ_{freq} are set to 10.0, 1.0 and 10.0, respectively. To optimize the model, we employ the Adam [45] optimizer with standard parameters. The initial learning rate is set to $1 \times 10^{-4.5}$ and is halved every 10K iterations to facilitate effective convergence and prevent overfitting.

Benchmarks. We demonstrate the validity of our approach by comparing with some sota image hiding methods, including ISN [3], HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37]. For fair comparison, we retrained the above five networks using the same dataset and settings.

Evaluation metrics. We assess the effectiveness of our method using four metrics: PSNR, SSIM [46], RMSE, and MAE. As shown below:

Algorithm 3: The training process of our model.

Input: The images x_{cov} and x_{sec} .
Output: Trained network I_Φ .

- 1 **Training Variable:** Θ_Φ .
- 2 Initialize Φ with Gaussian initialization;
- 3 **while** $step < max_steps$ **do**
- 4 | Compute $x_{ste} = \Phi(x_{cov}, x_{sec})$;
- 5 | Compute $x_{cov_rec}, x_{sec_rec} = \Phi(x_{ste}, z)$;
- 6 | Update $\Theta_\Phi \leftarrow \Theta_\Phi + lr \times Adam(L_{total})$;
- 7 **end**
- 8 **return** Θ_Φ

- **PSNR:** PSNR assesses the clarity and signal intensity of the reconstructed image by calculating the ratio of the highest pixel value to the noise level, typically quantified in decibels. The formulation of PSNR is as follows:

$$MSE = \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H [X_{i,j} - Y_{i,j}]^2, \quad (21)$$

TABLE III: Benchmark comparisons are presented for various datasets, showcasing the best results in bold and the second best are underlined. \uparrow indicates a preference for higher values, while \downarrow indicates the opposite.

| Methods | Cover/Stego image pair | | | | | | | | | | | |
|----------------|----------------------------|-----------------|------------------|-------------------|---------------------|-----------------|------------------|-------------------|---------------------|-----------------|------------------|-------------------|
| | DIV2K | | | | COCO | | | | ImageNet | | | |
| | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow |
| ISN [3] | 19.5486 | 0.5568 | 21.2815 | 27.8178 | 18.9135 | 0.5440 | 22.4384 | 29.7726 | 18.7818 | 0.5416 | 22.9882 | 30.4659 |
| HiNet [2] | 43.7374 | 0.9861 | 1.2324 | 1.8105 | 38.3738 | 0.9682 | 2.2369 | 3.3572 | 38.5275 | 0.9662 | 2.3277 | 3.3898 |
| IICNet [35] | 36.2106 | 0.9098 | 3.0764 | 3.9973 | 34.7038 | 0.9096 | 3.6146 | 4.7821 | 34.5877 | 0.9059 | 3.6924 | 4.8673 |
| DeepMIH [34] | <u>44.0757</u> | <u>0.9894</u> | <u>1.1786</u> | <u>1.7220</u> | <u>38.9650</u> | <u>0.9761</u> | <u>2.1125</u> | <u>3.1433</u> | <u>39.0314</u> | <u>0.9737</u> | <u>2.2176</u> | <u>3.2208</u> |
| iSCMIS [37] | 43.4168 | 0.9884 | 1.2459 | 1.8808 | 37.9841 | 0.9724 | 2.2976 | 3.5513 | 38.1072 | 0.9696 | 2.4162 | 3.6153 |
| DenseJIN(Ours) | 46.1608 | 0.9925 | 0.9506 | 1.3653 | 40.2858 | 0.9805 | 1.8317 | 2.7372 | 40.3525 | 0.9779 | 1.9438 | 2.8073 |
| Methods | Secret/Recovery image pair | | | | | | | | | | | |
| | DIV2K | | | | COCO | | | | ImageNet | | | |
| | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow |
| ISN [3] | 33.5509 | 0.9227 | 3.6890 | 5.8780 | 30.1825 | 0.8841 | 5.4381 | 8.6296 | 30.0229 | 0.8769 | 5.7234 | 8.9675 |
| HiNet [2] | 48.8182 | 0.9959 | 0.7195 | 1.0459 | 41.8110 | 0.9859 | 1.5144 | 2.3745 | 41.9348 | 0.9837 | 1.6097 | 2.4380 |
| IICNet [35] | 45.7237 | 0.9945 | 0.9374 | 1.4009 | <u>42.7490</u> | <u>0.9885</u> | <u>1.3343</u> | <u>2.1375</u> | 41.0416 | 0.9864 | 1.7229 | 2.5641 |
| DeepMIH [34] | <u>49.1999</u> | <u>0.9966</u> | <u>0.6862</u> | <u>0.9847</u> | 42.4502 | 0.9880 | 1.4343 | 2.1849 | 42.4641 | <u>0.9870</u> | 1.5467 | 2.3678 |
| iSCMIS [37] | 48.9661 | 0.9965 | 0.7055 | 1.0099 | 42.6928 | 0.9884 | 1.4056 | <u>2.1325</u> | <u>42.7719</u> | 0.9864 | <u>1.5055</u> | <u>2.2159</u> |
| DenseJIN(Ours) | 51.0916 | 0.9978 | 0.5525 | 0.7854 | 44.4273 | 0.9921 | 1.1582 | 1.7622 | 44.4513 | 0.9906 | 1.2408 | 1.8209 |

$$\text{PSNR} = 10 \cdot \log_{10} \frac{255^2}{\text{MSE}}, \quad (22)$$

where $X_{i,j}$ and $Y_{i,j}$ indicate the pixels at position (i, j) of images X and Y , respectively. The W indicates width and the H indicates height.

- SSIM: SSIM measures the similarity between images by analyzing the structural degradation of the images [46]. The formula is as follows:

$$\text{SSIM} = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \quad (23)$$

where the X and Y indicate the images; μ_X and μ_Y denote means; σ_X and σ_Y denote variances; and σ_{XY}^2 denotes covariances. c_1, c_2 are constants.

- MAE: MAE measures absolute error between two images, which is expressed as follows:

$$\text{MAE} = \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H |X_{i,j} - Y_{i,j}|, \quad (24)$$

- RMSE: RMSE is a common method for measuring the difference between two images, which is the square root of MSE (21) as shown below.

$$\text{RMSE} = \sqrt{\text{MSE}}. \quad (25)$$

A higher PSNR and SSIM, as well as lower RMSE and MAE, suggest superior image quality. As for image security, we leverage traditional steganalysis techniques and deep learning-powered steganalysis technique SRNet [47], ZhuNet [48] and SiaSteNet [49] to determine the detection rate. A detection rate that approaches 50%, representing random guessing, is indicative of a stronger security performance.

4.2 Our Results for Hiding a Single Image

Quantitative results. Table III presents a comparative analysis of our model's numerical results, along with those of other benchmark methods, focusing on metrics such as PSNR, SSIM, MAE, and RMSE. The evaluation is conducted on three datasets: DIV2K, COCO, and ImageNet. The table data unequivocally showcase the outstanding superiority of our model across all four metrics when evaluating cover/stego and secret/recovery image pairs. Regarding PSNR, our model exhibits a remarkable improvement over the second-best results on cover/stego image pairs across various datasets. Specifically, we observe a notable increase of 2.0851 dB on the DIV2K dataset, followed by gains of 1.3208 dB on the COCO dataset and 1.3211 dB on the ImageNet dataset. Moreover, when evaluating secret/recovery image pairs, our model surpasses the second-best results by 1.8917 dB, 1.6783 dB, and 1.6794 dB on the DIV2K, COCO, and ImageNet datasets, respectively. Furthermore, comparable enhancements are observed across metrics such as SSIM, RMSE, and MAE, in addition to PSNR. Thanks to the integration of dense connections and feature extraction modules in our model, it successfully mitigates feature loss, resulting in superior performance compared to the sota methods.

Qualitative results. Fig.5 illustrates a comparison between the image hiding and recovery outcomes obtained using our method and five other approaches. It is evident that our DenseJIN produces stego images with the most visually pleasing results, closely resembling the original image. In the stego images generated by the other five methods, noticeable artifacts related to texture replication can be observed. In contrast, our DenseJIN produces stego images with significantly reduced texture replication artifacts. Furthermore, when we compare the residual

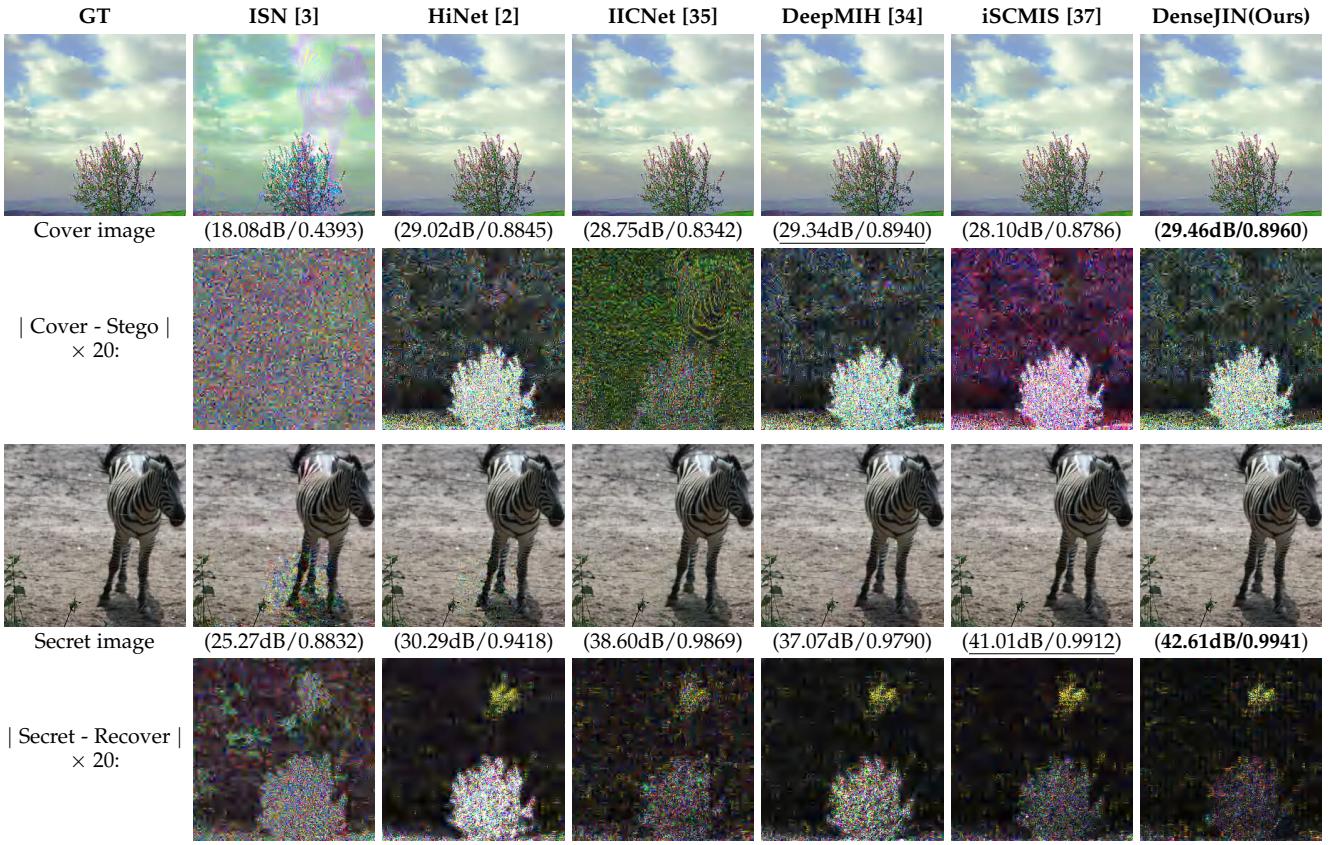


Fig. 5: Visual comparisons between the stego and recovery images produced by our DenseJIN model and the ISN [3], HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37] methods. The original cover image is represented in the first column of the first row, while the stego images generated with their corresponding models are depicted in the remaining six columns. The second row displays the residual map, indicating the variance between the cover image and the stego image. In the third row, the original secret image is showcased in the first column, with the subsequent six columns exhibiting the recovered secret images generated by their respective models. The fourth row presents the residual map, illustrating the disparity between the secret image and the recovered secret image.

maps of our method's stego images with the original cover images, they appear darker than those of the other five methods. The observed difference implies that DenseJIN demonstrates superior concealment capabilities in contrast to alternative methods. Among the recovered secret images, it is evident that the ISN [3] and HiNet [2] methods exhibit noticeable texture replication artifacts. While the recovered secret image of IICNet [35], DeepMIH [34] and iSCMIS [37] appear visually comparable to that of our method, the quantitative evaluation using PSNR/SSIM values indicates that our method outperforms IICNet [35], DeepMIH [34] and iSCMIS [37] in terms of similarity to the original secret image. Furthermore, the residual map between our method's recovered secret image and the original secret image appears darker than other five methods, indicating a more favorable recovery effect with our approach.

Embedded capacity. In our proposed DenseJIN, we embed a colour image of size (W, H) into another colour image of size (W, H) . Thus, the total effective number of bits is $W \times H \times 8 \times 3$ and the total number of pixels is $W \times H$. From this, we can calculate that the payload of our method to hide a colour image is 24 bpp. When we

embed only one colour image, the embedding capacity of our method is the same as ISN [3], HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37], but the embedding effect of our method is better than them.

4.3 Our Results for Hiding Multiple Images

Quantitative results. To evaluate the ability of the DenseJIN framework to conceal multiple images, we conducted experiments where 2-5 secret images were embedded within a single cover image. Table IV presents the average PSNR and SSIM values of the proposed method on the DIV2K dataset. As shown in Table IV, the concealing performance of the cover image degrades with an increasing number of secret images. This degradation is evident from the decreasing average PSNR values observed during the recovery process for the secret images. Our method demonstrates superior performance compared to the other five methods, especially when concealing 2 or 3 secret images, as shown in Table IV. Specifically, when concealing 2 secret images, our method achieves an impressive average PSNR value of 41.1157 dB for the cover/stego image pair, surpassing the second-place method, iSCMIS [37], by

TABLE IV: Benchmark comparisons on different datasets of hiding multiple images in one image, with the best results in bold and the second best are underlined. \uparrow indicates a preference for higher values, while \downarrow indicates the opposite.

| Methods | Cover/Stego image pair | | | | | | | |
|----------------|----------------------------|---------------|----------------|---------------|----------------|---------------|----------------|---------------|
| | 2 | | 3 | | 4 | | 5 | |
| | PSNR(dB) | SSIM | PSNR(dB) | SSIM | PSNR(dB) | SSIM | PSNR(dB) | SSIM |
| ISN [3] | 21.6165 | 0.5045 | 19.0564 | 0.4382 | 13.9287 | 0.2566 | 19.9785 | 0.4371 |
| HiNet [2] | 37.7684 | 0.9568 | 33.9112 | 0.9149 | 30.2340 | 0.8657 | <u>28.4440</u> | 0.8142 |
| IICNet [35] | 34.9362 | 0.8755 | 33.7553 | 0.8644 | 33.3001 | 0.8810 | 32.8402 | <u>0.8480</u> |
| DeepMIH [34] | 37.4325 | 0.9523 | <u>34.1871</u> | 0.9173 | 31.4850 | <u>0.9083</u> | 26.0836 | 0.6427 |
| iSCMIS [37] | <u>39.3815</u> | <u>0.9709</u> | 33.7045 | <u>0.9398</u> | 27.8127 | 0.8485 | 27.2191 | 0.8507 |
| DenseJIN(Ours) | 41.1157 | 0.9750 | 35.9073 | 0.9467 | 32.0624 | 0.9150 | 27.2529 | 0.8176 |
| Methods | Secret/Recovery image pair | | | | | | | |
| | 2 | | 3 | | 4 | | 5 | |
| | PSNR(dB) | SSIM | PSNR(dB) | SSIM | PSNR(dB) | SSIM | PSNR(dB) | SSIM |
| ISN [3] | 34.6227 | 0.9358 | 28.5142 | 0.8424 | 22.6666 | 0.6288 | 28.1235 | 0.8472 |
| HiNet [2] | 40.8761 | 0.9803 | 37.0136 | 0.9583 | 33.0946 | 0.9206 | 29.9829 | 0.9012 |
| IICNet [35] | 41.2023 | 0.8755 | 39.0545 | 0.9728 | 36.9767 | 0.9593 | 35.1979 | 0.9439 |
| DeepMIH [34] | 38.6234 | 0.9716 | 33.7605 | 0.9381 | 31.3713 | 0.9093 | 30.2369 | 0.9552 |
| iSCMIS [37] | <u>41.6296</u> | <u>0.9832</u> | 38.3800 | 0.9677 | <u>35.7628</u> | <u>0.9473</u> | <u>32.9492</u> | <u>0.9169</u> |
| DenseJIN(Ours) | 42.5745 | 0.9870 | 38.9202 | 0.9700 | 35.2657 | 0.9460 | 30.2821 | 0.9053 |

1.7342 dB. Additionally, the average PSNR value of the secret/recovery image pair reaches 42.5745 dB, which is 0.9449 dB higher than the second-place method, iSCMIS [37]. When the number of secret images exceeds 3, the performance of IICNet [35] exhibits a slight advantage over our method. This can be attributed to the inclusion of a relation module in IICNet [35] which introduces nonlinear transformations and enhances its ability to handle multiple secret images. The integration of such a module enables IICNet [35] to better adapt to scenarios involving a larger number of secret images, potentially explaining its superior performance in these cases.

Qualitative results. Fig.6 presents a comparison of our method with other techniques for concealing two color images of identical size within a single color image. Fig.6 clearly demonstrates that our method produces a stego image that closely resembles the original cover image. In contrast, the stego image produced by the ISN [3] exhibits noticeable color distortion and artifacts related to texture replication. Although the stego images produced by HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37] may exhibit more visually appealing results, it is important to note that the residual map between these stego images and the original cover image can raise suspicion. In contrast, the residual map between the stego image generated by our method and the original cover image predominantly appears black, indicating that our approach excels in effectively concealing multiple images. Comparing the secret images recovered by each method, it is evident that our approach yields a visually superior result for secret image 1 in comparison to the other five methods. The recovered secret image exhibits enhanced visual quality by using our method. Although the PSNR/SSIM values for the recovered secret image 2 and the original secret image

2 are slightly lower than those of Hinet [2] and IICNet [35], the perceptual difference is negligible. Our approach might have enabled the carrier image to conceal a greater amount of the information of secret image 1, which could have had a marginal effect on the restoration quality of secret image 2. Furthermore, upon comparing the residual maps between our method's generated secret image 2 and the original secret image, it is evident that our method's residual map contains the least information related to secret image 1.

Embedded capacity. In our method, the embedding capacity is 48-120 bpp when 2-5 colour images are embedded in a single colour image. The embedding capacity of ISN [3], HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37] is also 48-120 bpp, but from the data in Table IV, it can be found that our method has a great advantage when the embedding capacity is 48-72 bpp.

4.4 Visualisation of DenseJIN

Visualisation of the embedding process. Fig. 7 demonstrates the changes in the cover and secret branches for the forward embedding process of our method, and the subscripts LL, LH, HL and HH denote the corresponding wavelet subbands. From Fig. 7, it can be found that NIUM effectively extracts the contour feature information of the object in the secret image, and embedding this information into the cover image helps to extract the high quality secret image. In addition, the colour of the image of the cover branch gradually becomes darker from INN1 to INN7, while the quality of the image of the output cover branch of INN8 is excellent, which proves the effectiveness of our proposed dense connection. From the wavelet subbands corresponding to the cover and secret branches, it can be found that the information of the cover image and the secret

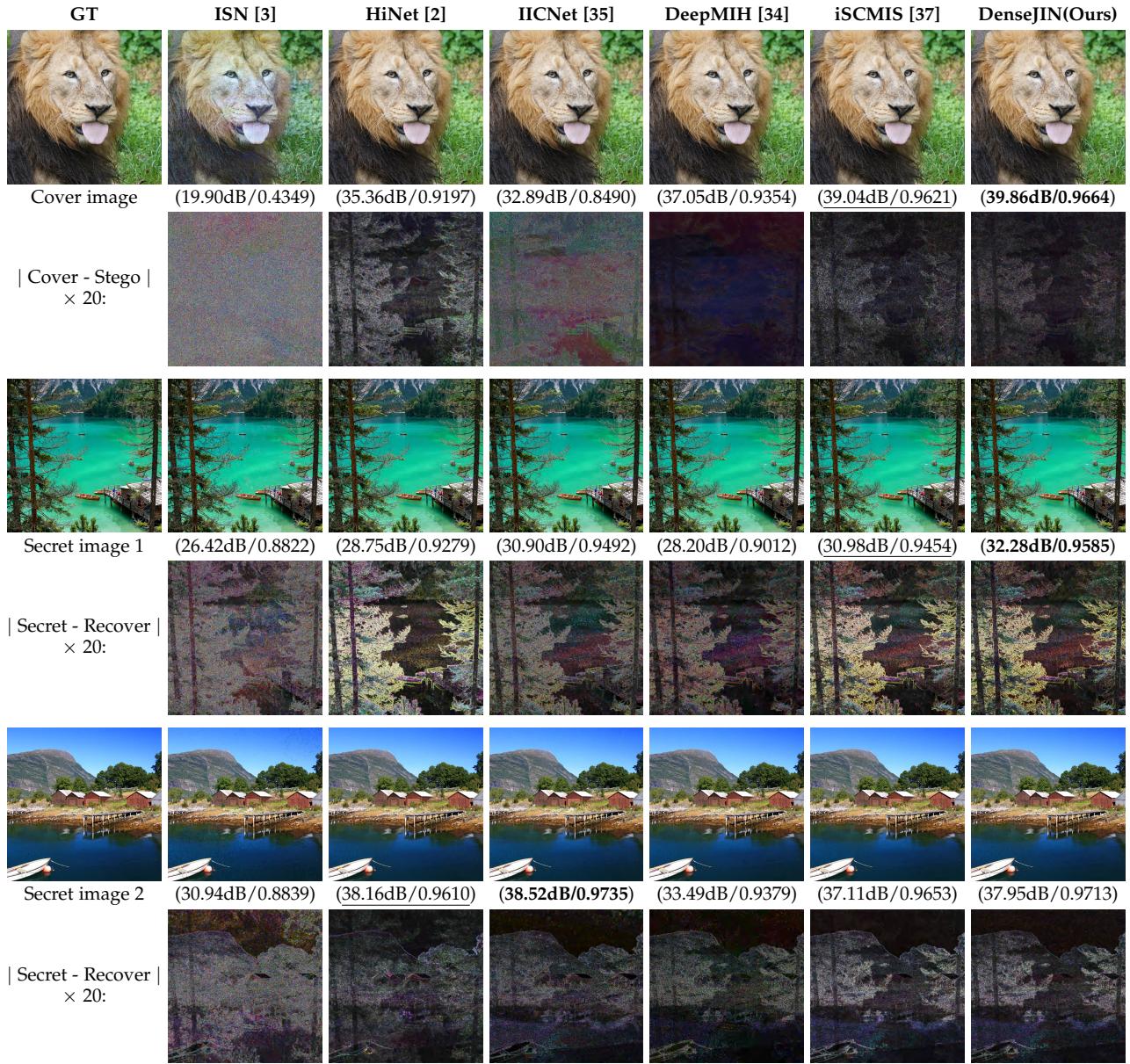


Fig. 6: Visual comparisons of stego and recovery images are shown when we hide 2 secret images in one cover image with our DenseJIN and the comparison methods. In the first row, the original cover image is represented in the first column, while the stego images generated by each respective model are displayed in the remaining six columns. The second row illustrates the residual map between the cover image and the corresponding stego image. Transitioning to the third row, the original secret image 1 is presented in the first column, with the recovered secret image 1 produced by each model exhibited in the remaining six columns. Following the same sequence, the fourth row illustrates the residual map between the secret image and the recovered secret image for secret image 1. Continuing in this manner, the fifth and sixth rows display the residual map and recovered secret image for secret image 2.

image are fused with each other after the affine coupling computation, and the secret image is gradually embedded into the cover image after a number of invertible blocks.

Visualisation of the extracting process. Fig. 8 shows the changes of the cover and secret branch images during the backward extraction process of our method. The image of the coover branch is gradually identical to the original cover image after the affine coupling computation of several invertible blocks, and the image of the secret branch is

gradually transformed from a matrix randomly sampled from a Gaussian distribution to the secret image. From the wavelet subbands corresponding to the cover and secret branches, it can be found that even if the input of the secret branch is a random matrix, the invertible blocks can extract the information of the secret image from the secret image through the affine coupling computation, and finally recover the secret image.

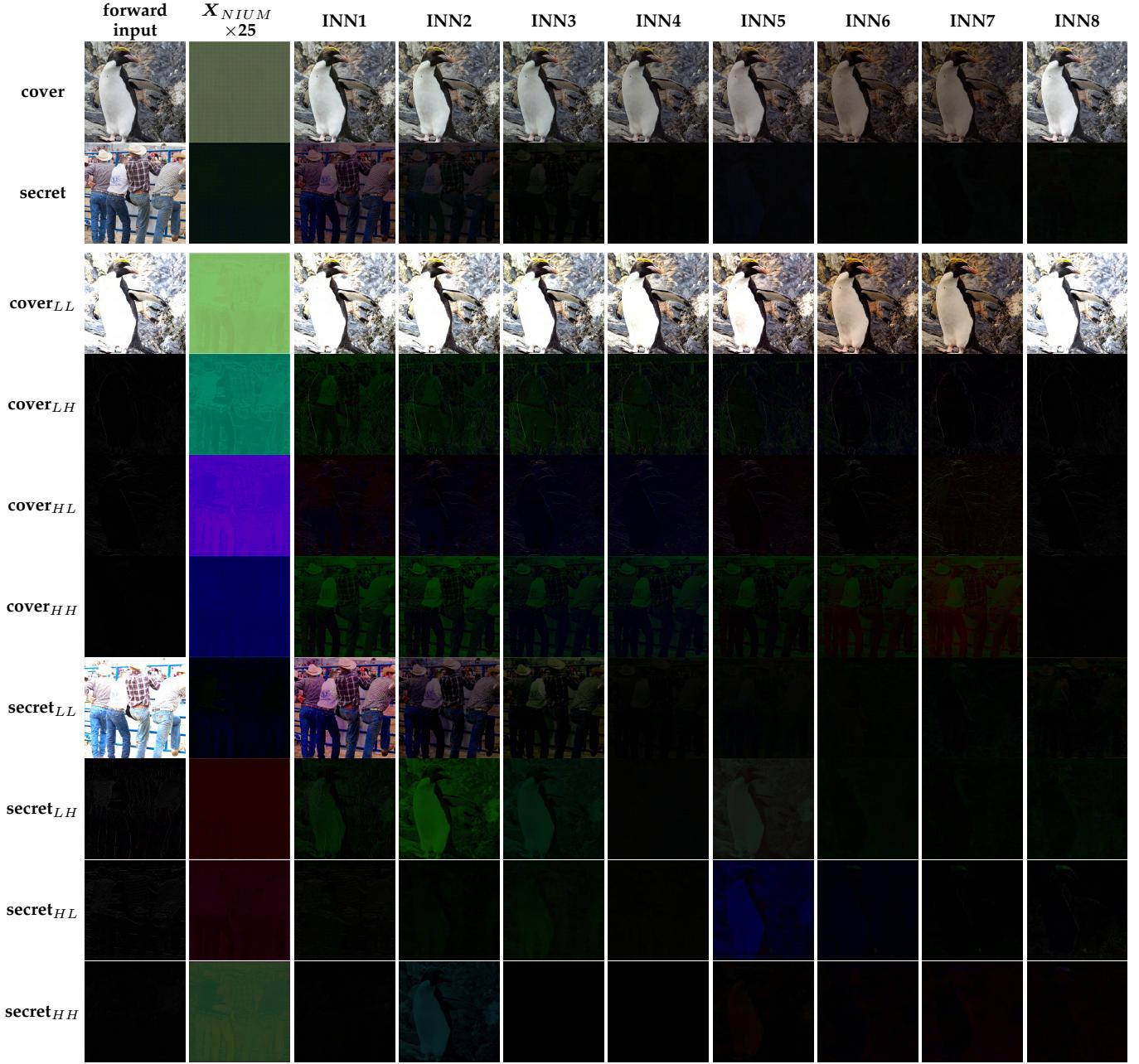


Fig. 7: The visualization of the forward embedding process of our model. In this figure, the outputs of the NIUM and each invertible block are shown. INN1 to INN8 denote the forward outputs of the 1st to 8th invertible block, and $X_{NIUM} \times 25$ denotes the outputs of NIUM multiplied by a factor of 25.

4.5 Steganographic Analysis

Steganographic analysis plays a crucial role in evaluating the security of stego images, making itself an essential aspect of the image hiding task. Primarily, steganalysis assesses the ability to differentiate stego images from cover images by using specialized steganalysis tools [50]. We conduct experiments using both traditional statistical steganalysis approaches and deep learning-based steganalysis approaches to demonstrate the effectiveness of our model in resisting steganalysis.

Traditional statistical steganalysis. In general, a histogram can represent the distribution of pixel values

in an image. Based on the histogram it is possible to determine the extent to which the original cover image has been modified. Fig. 9 shows the histograms of the stego image obtained by the various methods in Fig. 5. Fig. 9 illustrates that the alteration of the cover image by ISN [3] is substantial. HiNet [2], IICNet [35], DeepMIH [34], iSCMIS [37] and our method fluctuates less in the pixel value modification of the cover image, which can be considered to be caused by adding noise to the cover image. Therefore, our method can evade the histogram distortion detection.

In line with established practices [51], we employed StegExpose [52] to assess DenseJIN's resistance to

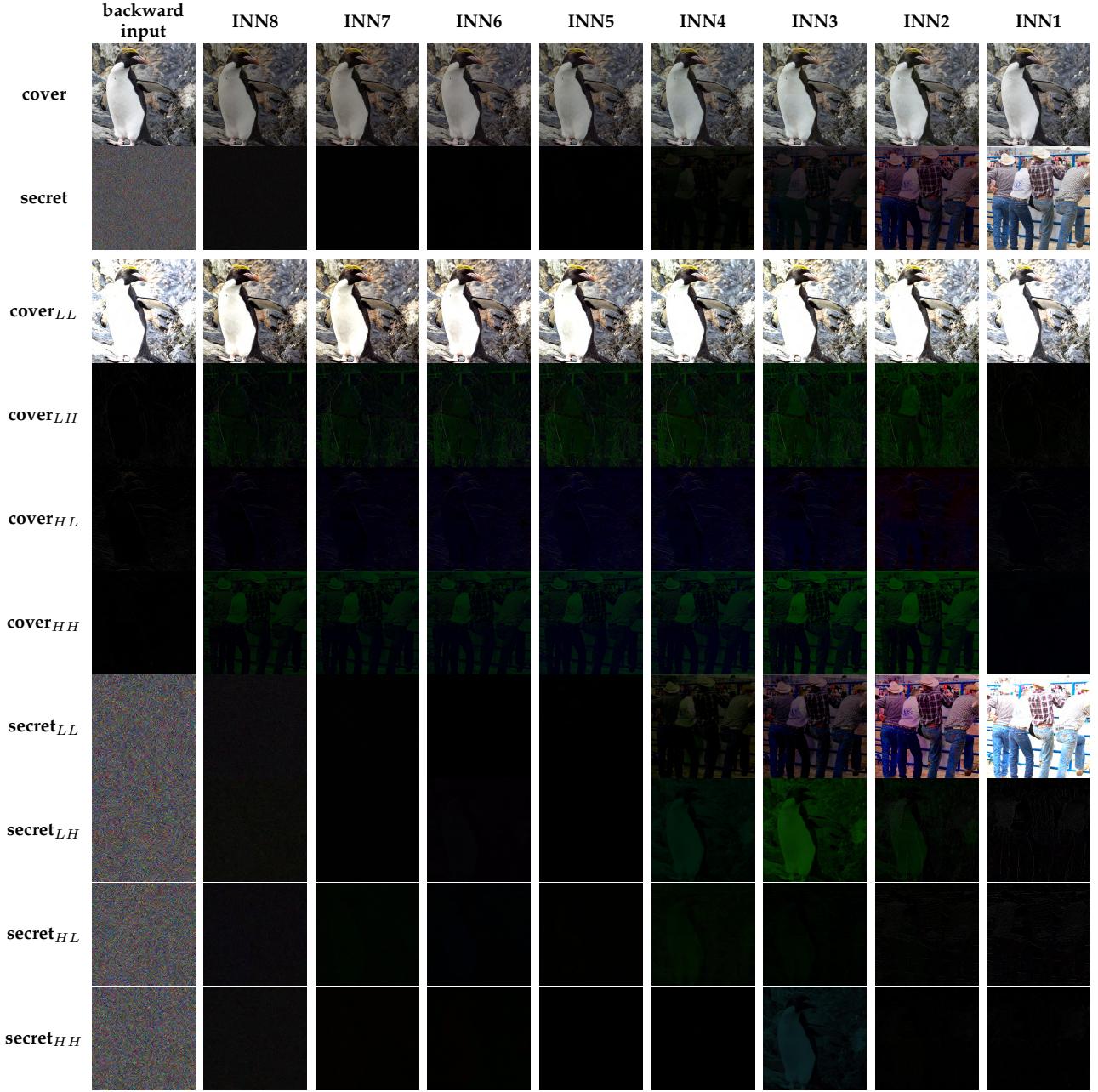


Fig. 8: The visualization of our model extracting secret images. In this figure, the outputs of and each invertible block are shown. INN1 to INN8 denote the backward outputs of the 1st to 8th invertible block.

steganalysis. To conduct the evaluation, we randomly selected 1,600 images from the test dataset, with 800 images serving as cover images and the remaining 800 images as secret images. Then, we employed our model to generate stego images based on this selection. Subsequently, we employed our model to recover the secret images from the generated stego images. Notably, a detection accuracy approaching 50% indicates a stronger resistance to steganalysis. We employed StegExpose [52] to detect the stego images generated by our method and other methods, and the resulting ROC curve is illustrated in Fig. 10. The AUC of DenseJIN is calculated to be 0.5562, which is only 0.0017 higher than that of iSCMIS [37], and the difference between the detected rate of our method and iSCMIS [37]

is very small. This value can indicate that the detection accuracy is very close to the accuracy of random guessing. Our model's ability to produce stego images with a high level of security is clearly evident in the results, which indicates a significantly higher probability of evading detection by the StegExpose tool.

Deep learning-based steganalysis. We use three sota steganalysis networks SRNet [47], ZhuNet [48] and SiaSteNet [49] to detect whether the stego image contains secret information. To be fair, we retrain these three steganalysis networks using the results obtained by each steganography model on the DIV2K test set. Tables V, VI and VII show the detection rates of different image

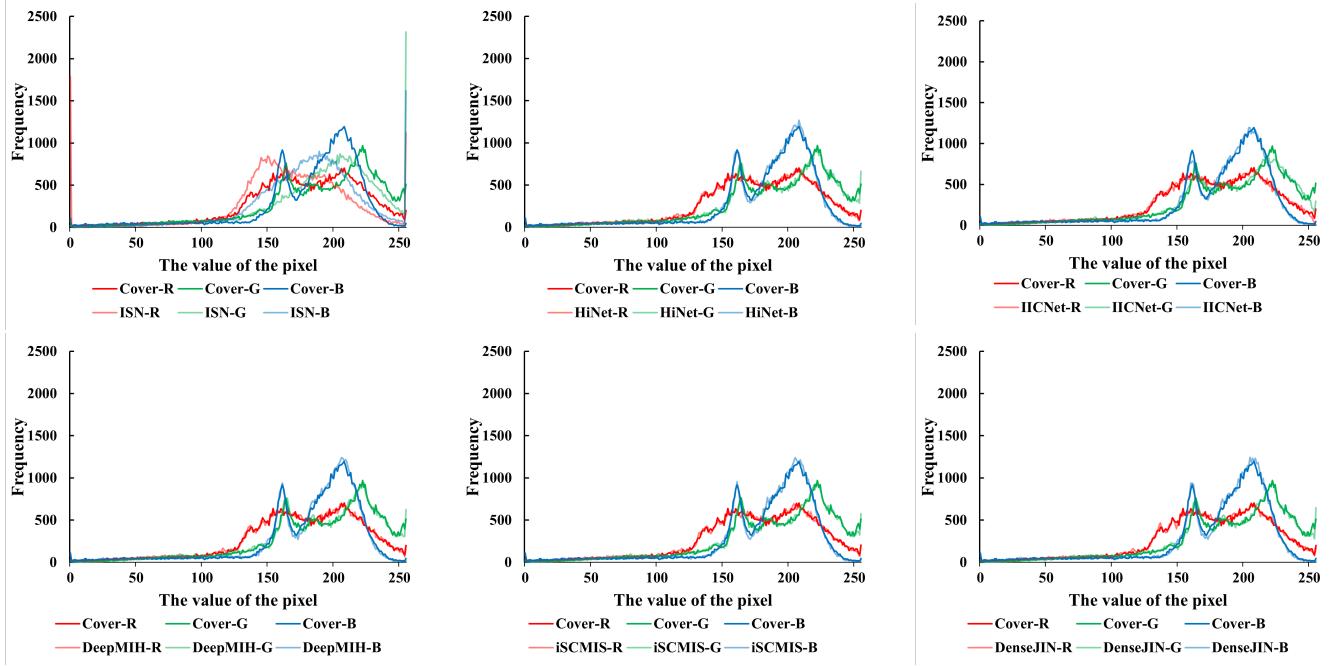


Fig. 9: Comparison of our method with ISN [3], HiNet [2], IICNet [35], DeepMIH [34] and iSCMIS [37] in terms of histogram distortion, including histograms of the three RGB channels. The darker colour is the histogram of the cover image and the lighter colour is the histogram of the stego image generated by the corresponding method.

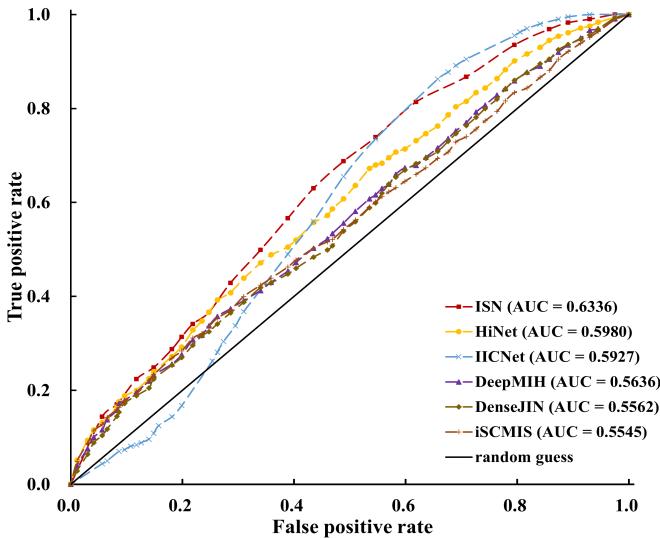


Fig. 10: The ROC curve produced by StegExpose.

steganography methods on SRNet [47], ZhuNet [48] and SiaSteNet [49] respectively. The closer the detection rate is to 50% indicates that the method has higher resistance to steganalysis. In Table V, the detection rate of our DenseJIN is closer to 50% than other methods, reaching 56.82%, which indicates that our method is able to spoof SRNet [47]. For ZhuNet [48], although the detection rate of our DenseJIN is not the closest to 50%, the detection rate is at a low level, which indicates that our method is able to effectively resist the detection of ZhuNet [48]. For SiaSteNet [49], the detection rates of all methods except IICNet [35] are over

TABLE. V: The detection accuracy achieved by using SRNet [47].

| Methods | Accuracy (%) \pm std |
|----------------|-------------------------|
| ISN [3] | 86.68 \pm 1.13 |
| HiNet [2] | 86.80 \pm 1.71 |
| IICNet [35] | 94.74 \pm 0.34 |
| DeepMIH [34] | 80.56 \pm 2.39 |
| iSCMIS [37] | 75.91 \pm 2.36 |
| DenseJIN(Ours) | 56.82 \pm 3.40 |

TABLE. VI: The detection accuracy achieved by using ZhuNet [48].

| Methods | Accuracy (%) \pm std |
|----------------|-------------------------|
| ISN [3] | 73.00 \pm 1.56 |
| HiNet [2] | 66.53 \pm 0.79 |
| IICNet [35] | 57.37 \pm 0.21 |
| DeepMIH [34] | 59.99 \pm 0.39 |
| iSCMIS [37] | 61.12 \pm 0.63 |
| DenseJIN(Ours) | 66.87 \pm 1.12 |

90%, which may be due to the fact that IICNet [35] adds a relation module that allows it to better capture the relations of the cross-images. However, the detection rate of our DenseJIN is lower than the other three methods, indicating that our method has a greater advantage.

In addition to the aforementioned steganalysis methods, a novel technique for steganalysis was proposed by Weng et al. [40] who retrained steganalysis network using a specific model's cover/stego image pairs. They systematically varied the number of training images to investigate the min-

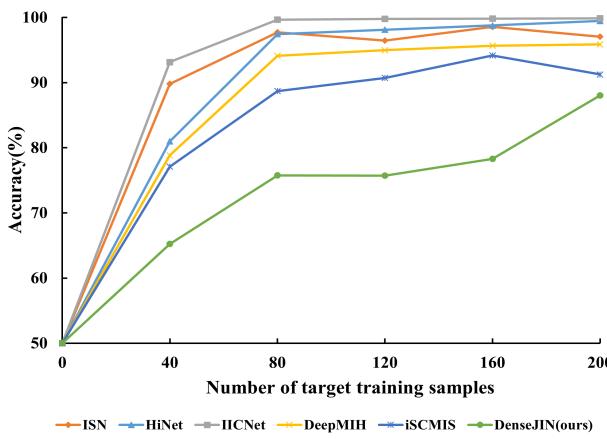


Fig. 11: Investigation of the resistance to SRNet [47] of different approaches.

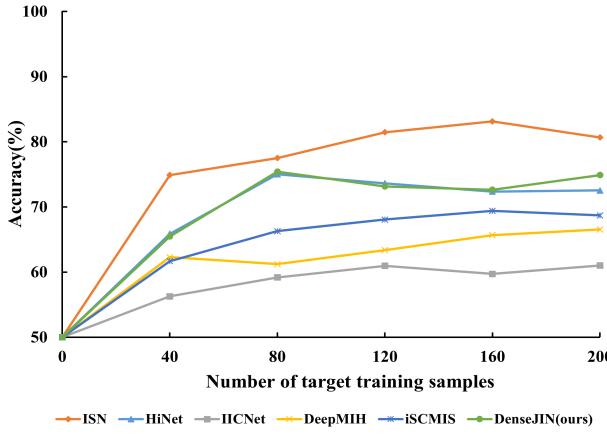


Fig. 12: Investigation of the resistance to ZhuNet [48] of different approaches.

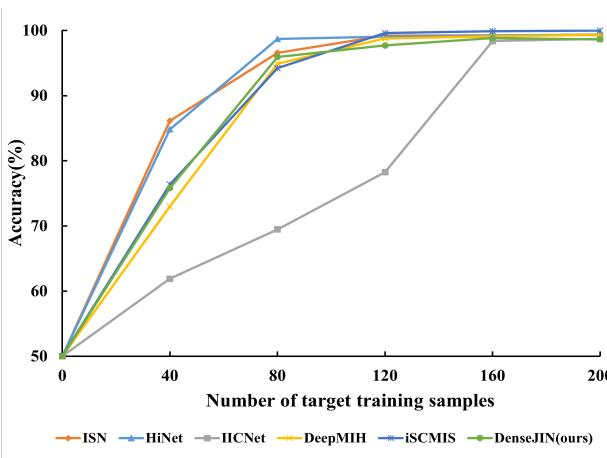


Fig. 13: Investigation of the resistance to SiaSteNet [49] of different approaches.

TABLE. VII: The detection accuracy achieved by using SiaSteNet [49].

| Methods | Accuracy (%) \pm std |
|----------------|-------------------------|
| ISN [3] | 93.20 \pm 0.19 |
| HiNet [2] | 93.48 \pm 0.28 |
| IICNet [35] | 75.05 \pm 1.53 |
| DeepMIH [34] | <u>90.44</u> \pm 0.12 |
| iSCMIS [37] | 91.10 \pm 0.25 |
| DenseJIN(Ours) | 90.51 \pm 0.09 |

imum quantity required for steganalysis network to achieve reliable stego image detection capabilities. Following the procedure outlined in [40], we incrementally increased the number of samples used to train SRNet [47], ZhuNet [48] and SiaSteNet [49]. Fig. 11, 12 and 13 depict the change in detection accuracy as the number of training samples increase. For SRNet [47], our DenseJIN consistently achieved significantly lower detection accuracy in comparison to other approaches. For ZhuNet [48], our DenseJIN is not the best, but the detection rate is consistently below 75%. It can be found from Table. VII that our DenseJIN is second only to IICNet [35] in SiaSteNet [49]. The above results show that our DenseJIN has a comparative advantage over other methods in resisting deep learning-based steganalysis.

4.6 Ablation Study

Effectiveness of dense connections. Table VIII provides compelling evidence of the vital role played by dense connections in enhancing the effectiveness of our method. Specifically, regarding image embedding, the incorporation of dense connections leads to a noteworthy average increase of 2.5183 dB in PSNR values for stego images. Furthermore, for image recovery, the inclusion of dense connections results in an improvement of 1.6877 dB in PSNR values for the extracted images. These findings strongly affirm the effectiveness of dense connections in facilitating image hiding, underscoring their indispensable contribution to the overall performance enhancement of our method.

Effectiveness of the noninvertible block. Table VIII clearly demonstrates the significant improvements achieved by the noninvertible block in the embedding and extracting processes. Specifically, when we employ the noninvertible block during the embedding process, there is a notable enhancement of 0.7780 dB in PSNR values for stego images. Similarly, incorporating the noninvertible block results in a substantial average increase of 0.4250 dB in PSNR values for the extracted images during the extraction process. These improvements are further shown by the other three metrics, highlighting the effectiveness of the noninvertible block in enhancing both the concealing and revealing results.

4.7 Computational Complexity Analysis

In this section, we measure the time complexity of different image steganography models based on INNs, and the results are presented in Table IX. All the experiments are implemented on Nvidia A30 GPUs. From Table IX, we can find that ISN [3] has the least number of parameters because

TABLE. VIII: Ablation study of dense connections and the noninvertible block. The following data are the results on the DIV2K [42] test set. \uparrow indicates a preference for higher values, while \downarrow indicates the opposite. The detection rate is the result of the detection using SRNet [47].

| Dense connect | Noninvertible block | Cover/Stego image pair | | | | Secret/Recovery image pair | | | | Detection rate(%) |
|---------------|---------------------|------------------------|-----------------|------------------|-------------------|----------------------------|-----------------|------------------|-------------------|-------------------|
| | | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | PSNR(dB) \uparrow | SSIM \uparrow | MAE \downarrow | RMSE \downarrow | |
| \times | \times | 44.0757 | 0.9894 | 1.1786 | 1.7220 | 49.1999 | 0.9965 | 0.6862 | 0.9847 | 78.15 |
| \times | \checkmark | 44.8537 | 0.9914 | 1.0697 | 1.6372 | 49.6249 | 0.9971 | 0.6562 | 0.9490 | 73.53 |
| \checkmark | \times | 46.5940 | 0.9929 | 0.9072 | 1.2951 | 50.8876 | 0.9977 | 0.5662 | 0.8044 | 61.89 |
| \checkmark | \checkmark | 46.1608 | 0.9925 | 0.9506 | 1.3653 | 51.0916 | 0.9978 | 0.5525 | 0.7854 | 56.82 |

ISN [3] simply stacks a number of invertible blocks. Our DenseJIN has the highest number of parameters because we added an extra Unet module to the model for feature extraction. Although we have the most parameters, we do not have the longest training time or testing time because the Unet and the dense connect are only involved in the embedding process of the secret image. This also shows that our DenseJIN has good performance in terms of time complexity.

TABLE. IX: Performance comparison of our method with other sota methods. Training time refers to the time it takes to train an iteration. Testing time refers to the time required to hide one image.

| Methods | Parameter(M) | Training(s) | Testing(s) |
|----------------|--------------|-------------|------------|
| ISN [3] | 1.49 | 53 | 14 |
| HiNet [2] | 2.03 | 57 | 8 |
| IICNet [35] | 2.90 | 66 | 14 |
| DeepMIH [34] | 2.70 | 72 | 11 |
| iSCMIS [37] | 3.02 | 44 | 10 |
| DenseJIN(Ours) | 11.57 | 60 | 9 |

5 CONCLUSION AND FUTURE WORK

In this paper, we introduce the DenseJIN model to learn a joint representation for the concealment and recovery of secret images, which significantly improves the performance of sota methods. To be more specific, in the forward hiding process, we densely connect the invertible blocks in the INNs to minimize information loss. Additionally, we adopt a modified Unet to extract deep and fine-grained information, which helps to restore high-quality secret images. Extensive experiments conducted on the ImageNet, COCO, and DIV2K datasets demonstrate the high capacity and invisibility of our method. The results showcase its superiority over other sota methods in both qualitative and quantitative evaluations. Our proposed DenseJIN can be applied to covert transmission scenarios in lossless environments. For future research, we are interested in further investigating image steganography under noise and JPEG interference.

REFERENCES

- [1] Y. J. Chanu, K. M. Singh, and T. Tuithung, "Image steganography and steganalysis: A survey," *International Journal of Computer Applications*, vol. 52, no. 2, 2012.
- [2] J. Jing, X. Deng, M. Xu, J. Wang, and Z. Guan, "Hinet: Deep image hiding by invertible network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4733–4742.
- [3] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, "Large-capacity image steganography based on invertible neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10816–10825.
- [4] S. Katzenbeisser and F. A. Petitcolas, "Defining security in steganographic systems," in *Security and Watermarking of Multimedia Contents IV*, vol. 4675. SPIE, 2002, pp. 50–56.
- [5] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.
- [6] A. A. Tamimi, A. M. Abdalla, and O. Al-Allaf, "Hiding an image inside another image using variable-rate steganography," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 4, no. 10, 2013.
- [7] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE security & privacy*, vol. 1, no. 3, pp. 32–44, 2003.
- [8] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 685–696, 2020.
- [9] W. Su, J. Ni, X. Hu, and J. Fridrich, "Image steganography with symmetric embedding using gaussian markov random field model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1001–1015, 2021.
- [10] J. Tao, S. Li, X. Zhang, and Z. Wang, "Towards robust image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 2, pp. 594–600, 2019.
- [11] L. Liu, L. Meng, Y. Peng, and X. Wang, "A data hiding scheme based on u-net and wavelet transform," *Knowledge-Based Systems*, vol. 223, p. 107022, 2021.
- [12] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a u-net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.
- [13] Y. Wang, M. Xiao, C. Liu, S. Zheng, and T.-Y. Liu, "Modeling lost information in lossy image compression," *arXiv preprint arXiv:2006.11999*, 2020.
- [14] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, "Guided image generation with conditional invertible neural networks," *arXiv preprint arXiv:1907.02392*, 2019.
- [15] M. Xiao, S. Zheng, C. Liu, Y. Wang, D. He, G. Ke, J. Bian, Z. Lin, and T.-Y. Liu, "Invertible image rescaling," in *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 126–144.
- [16] C. Xia, Q. Guan, X. Zhao, and K. Wu, "Improved jpeg phase-aware steganalysis features using multiple filter sizes and difference images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4100–4113, 2020.
- [17] Q. Guan, K. Chen, H. Chen, W. Zhang, and N. Yu, "Detecting steganography in jpeg images recompressed with the same quantization matrix," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 6002–6016, 2022.
- [18] P. Tsai, Y.-C. Hu, and H.-L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal processing*, vol. 89, no. 6, pp. 1129–1143, 2009.
- [19] F. Pan, J. Li, and X. Yang, "Image steganography method based on pvd and modulus function," in *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, 2011, pp. 282–284.

- [20] S. Imaizumi and K. Ozawa, "Multibit embedding algorithm for steganography of palette-based images," in *Image and Video Technology: 6th Pacific-Rim Symposium, PSIVT 2013, Guanajuato, Mexico, October 28–November 1, 2013. Proceedings 6*. Springer, 2014, pp. 99–110.
- [21] M. Niimi, H. Noda, E. Kawaguchi, and R. O. Eason, "High capacity and secure digital steganography to palette-based images," in *Proceedings. International Conference on Image Processing*, vol. 2. IEEE, 2002, pp. II–II.
- [22] J. Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3. IEEE, 1996, pp. 239–242.
- [23] Z. Dawei, C. Guanrong, and L. Wenbo, "A chaos-based robust wavelet-domain watermarking algorithm," *Chaos, Solitons & Fractals*, vol. 22, no. 1, pp. 47–54, 2004.
- [24] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on image processing*, vol. 8, no. 1, pp. 58–68, 1999.
- [25] S. Baluja, "Hiding images in plain sight: Deep steganography," *Advances in neural information processing systems*, vol. 30, 2017.
- [26] K. A. Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "Steganogan: High capacity image steganography with gans," *arXiv preprint arXiv:1901.03892*, 2019.
- [27] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.
- [28] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.
- [29] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Advances in neural information processing systems*, vol. 31, 2018.
- [30] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte, "Srfflow: Learning the super-resolution space with normalizing flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 715–732.
- [31] Y. Xie, K. L. Cheng, and Q. Chen, "Enhanced invertible encoding for learned image compression," in *Proceedings of the 29th ACM international conference on multimedia*, 2021, pp. 162–170.
- [32] Y. Liu, Z. Qin, S. Anwar, P. Ji, D. Kim, S. Caldwell, and T. Gedeon, "Invertible denoising network: A light solution for real noise removal," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 365–13 374.
- [33] L. Zhao, S.-P. Lu, T. Chen, Z. Yang, and A. Shamir, "Deep symmetric network for underexposed image enhancement with recurrent attentional learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 075–12 084.
- [34] Z. Guan, J. Jing, X. Deng, M. Xu, L. Jiang, Z. Zhang, and Y. Li, "Deepmih: Deep invertible network for multiple image hiding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 372–390, 2022.
- [35] K. L. Cheng, Y. Xie, and Q. Chen, "Iicnet: A generic framework for reversible image conversion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1991–2000.
- [36] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7875–7884.
- [37] F. Li, Y. Sheng, X. Zhang, and C. Qin, "iscmis: Spatial-channel attention based deep invertible network for multi-image steganography," *IEEE Transactions on Multimedia*, 2023.
- [38] C. Mou, Y. Xu, J. Song, C. Zhao, B. Ghanem, and J. Zhang, "Large-capacity and flexible video steganography via invertible neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 22 606–22 615.
- [39] X. Hu, Z. Fu, X. Zhang, and Y. Chen, "Invisible and steganalysis-resistant deep image hiding based on one-way adversarial invertible networks," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2023.
- [40] X. Weng, Y. Li, L. Chi, and Y. Mu, "High-capacity convolutional video steganography with temporal residual modeling," in *Proceedings of the 2019 on international conference on multimedia retrieval*, 2019, pp. 87–95.
- [41] J. Fridrich, M. Goljan, and R. Du, "Detecting lsb steganography in color, and gray-scale images," *IEEE multimedia*, vol. 8, no. 4, pp. 22–28, 2001.
- [42] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126–135.
- [43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, pp. 211–252, 2015.
- [45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [47] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.
- [48] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial cnn-based steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1138–1150, 2019.
- [49] W. You, H. Zhang, and X. Zhao, "A siamese cnn for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 291–306, 2020.
- [50] X. Zhang, X. Zhang, and G. Feng, "Image steganalysis network based on dual-attention mechanism," *IEEE Signal Processing Letters*, 2023.
- [51] S. Baluja, "Hiding images within images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 7, pp. 1685–1697, 2019.
- [52] B. Boehm, "StegExpose-a tool for detecting lsb steganography," *arXiv preprint arXiv:1410.6656*, 2014.

Delin Duan received his Bachelor Degree from the School of Software, Jiangxi Agricultural University, NanChang, China, in 2020. He is studying for a Master Degree in School of Software of South China Normal University, FoShan, China. His research interests include image processing, information security, image steganography, and etc.



Shuyuan Shen received her M.S degree in Applied Mathematics from School of Mathematical Sciences from South China Normal University, Guangzhou, China, in 2005. She received Ph.D. degree in Applied Mathematics at the college of Mathematics and Econometrics, Hunan University, Changsha, China, 2015. Since July 2005, she has been in South China Normal University, Guangzhou, China. Her research interests include differential equation, data hiding, image processing and neural network.





SongSen Yu received the B.S. degree in electrical engineering in 1990 and the Ph.D. Degree in automatic control from Guangdong University of Technology, Guangzhou, China, in 2006. In July 1994, He joined the School of Computer at NanChang University, initially as a Teaching Assistant, a lecturer and subsequently as an Associate Professor. He was a Research Assistant from 2006 to 2008 in the School of Technology at SUN YAT-SEN University. Since 2009, he has been in School of Computer at the South China Normal University and is currently a full professor from 2012. His research interests are in the areas of Internet of things, RFID anti-collision, RFID security and wireless sensor networks. He has more than 30 publications in the above areas and is actively taking charge or participating in several R&D projects of the China.



Yibo Yuan received his Bachelor Degree from the the School of Energy and Mechanical Engineering, Jiangxi University of Science and Technology, NanChang, China, in 2021. He is studying for a Master Degree in School of Software of South China Normal University, FoShan, China. His research interests include image processing, information security, image watermarking, and etc.



Qidong Zhou received his Bachelor Degree from the the School of Software of South China Normal University, FoShan, China, in 2022. He is studying for a Master Degree in School of Software of South China Normal University, FoShan, China. His research interests include image processing, information security, image watermarking, and etc.



Haojie Lu is studying for a Master Degree in School of Software of South China Normal University, FoShan, China. His research direction is digital watermarking and information hiding.



Huanjie Lin received his Bachelor Degree from the School of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou, China, in 2020. He is studying for a Master Degree in School of Software of South China Normal University, FoShan, China. His research interests include image processing, information security, image watermarking, and etc.