HiNet: Deep Image Hiding by Invertible Network

Junpeng Jing[1,*], Xin Deng[1,*], Mai Xu[2,†], Jianyi Wang[2], Zhenyu Guan[1,†]
[1]School of Cyber Science and Technology, Beihang University, Beijing, China
[2]School of Electronic and Information Engineering, Beihang University, Beijing, China
{junpengjing, cindydeng, MaiXu, guanzhenyu}@buaa.edu.cn

Abstract

Image hiding aims to hide a secret image into a cover image in an imperceptible way, and then recover the secret image perfectly at the receiver end. Capacity, invisibility and security are three primary challenges in image hiding task. This paper proposes a novel invertible neural network (INN) based framework, HiNet, to simultaneously overcome the three challenges in image hiding. For large capacity, we propose an inverse learning mechanism by simultaneously learning the image concealing and revealing processes. Our method is able to achieve the concealing of a full-size secret image into a cover image with the same size. For high invisibility, instead of pixel domain hiding, we propose to hide the secret information in wavelet domain. Furthermore, we propose a new low-frequency wavelet loss to constrain that secret information is hidden in high-frequency wavelet subbands, which significantly improves the hiding security. Experimental results show that our HiNet significantly outperforms other state-of-the-art image hiding methods, with more than 10 dB PSNR improvement in secret image recovery on ImageNet, COCO and DIV2K datasets. Codes are available at https://github.com/TomTomTommi/HiNet.

1. Introduction

The task of image hiding is to conceal a secret image into a cover image to generate a stego image, which only allows the informed receivers to recover the secret image, but invisible to other people. For security concern, the stego image is usually required to be indistinguishable from the cover image. Different from bit-level message hiding or steganography [2, 20, 35, 36, 39–41], image hiding is more challenging, which requires large capacity, high invisibility and security. Image hiding has a wide range of applications, of which secret communication and privacy protection are the most significant ones. Compared to the well-known image cryptography, image hiding has a remarkable security advantage, i.e., the stego image with secret information inside is indistinguishable from the cover image, which makes it more suitable for secret communication. In addition, unlike image cryptography, image hiding focuses more on the capacity and invisibility of hidden information rather than robustness.

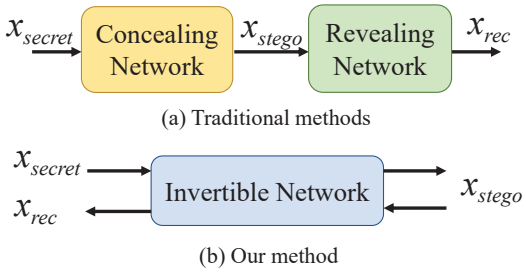

Figure 1. The illustration of difference between our image hiding method and the traditional methods [5, 23, 32].

Traditional steganographic approaches can only hide a small amount of information [6, 11, 13, 16, 19, 24], which cannot meet the requirement of large capacity in image hiding task. Baluja [4] proposed the first convolutional neural network (CNN) to solve image hiding problem. This work was then extended in [5] by permuting the pixels of secret image to enhance the hiding security. Weng *et al.* [32] further proposed a deep network for video steganography by temporal residual modeling. However, all these methods adopt two sub-networks for image hiding: a concealing network to hide a secret image $\boldsymbol{x}_{\text{secret}}$ into a cover image to generate a stego image $\boldsymbol{x}_{\text{stego}}$, and a revealing network to recover the secret image $\boldsymbol{x}_{\text{rec}}$ from $\boldsymbol{x}_{\text{stego}}$, as shown in Fig. 1 (a). The concealing and revealing networks have two sets of parameters, which are linked through simple concatenation. This loose connection may cause color distortion and texture-copying artifacts. Besides, they barely consider the security issue, making hidden secret information easy to be detected.

In this paper, we propose an invertible image hiding network, HiNet, in which the concealing and revealing pro-

*Authors contributed equally.
†Corresponding author.

cesses share the same set of network parameters, as shown in Fig. 1 (b). To the best of our knowledge, our work is the first attempt to explore invertible network in image hiding task. The main novelty is that image revealing is modelled as the reverse process of image concealing in an invertible network architecture, which means the network only needs to be trained once to get all network parameters for both concealing and revealing. This is a radical difference from the existing methods [5, 23, 32] which treat the concealing and revealing processes independently. Consequently, our HiNet achieves state-of-the-art performance on recovery accuracy, hiding security and invisibility. The main contributions of this paper are summarized as follows:

- We propose a novel image hiding network, namely HiNet, based on invertible neural network for the task of large-capacity image hiding.
- We design two concealing and revealing modules with differentiable and invertible property, aiming to make the image hiding process fully reversible.
- We propose a low-frequency wavelet loss to control the distribution of secret information in different frequency bands, which significantly improves the hiding security.

2. Related Work

2.1. Steganography and Image Hiding

Steganography is the practice of hiding one message, audio, image or video into another, in a way that does not arouse any suspicion. Least Significant Bit (LSB) [26] is the most traditional spatial domain based method in steganography. It works by replacing the n least significant bits of cover image with the most significant n bits of secret image. The disadvantage of LSB algorithm is the texture-copying artifacts, which often appear in smooth regions in an image. Thus, the steganalysis methods [11, 16, 19] can easily detect the existence of secret information hidden by LSB. In addition to LSB, there are many methods proposed to embed information in frequency domains, such as discrete Fourier transform (DFT) domain [24], discrete cosine transform (DCT) domain [13], and discrete wavelet transform (DWT) domain [6]. These methods are more robust and undetectable than LSB, but they can only hide bit-level information.

Recently, some deep learning models [2, 12, 20, 34–37, 39–41] have been proposed for steganography, which achieved better performance than traditional methods. Specifically, Zhu *et al.* [41] firstly proposed a network based on auto-encoder to achieve watermark embedding and extracting. Based on [41], Ahmadi *et al.* [2] introduced residual connections and a CNN-based transform operation module to embed watermarking in any transform space. Tancik *et al.* [27] proposed a StegaStamp framework to hide hyperlinks in a physical photograph and successfully retrieve it after decoding. Luo *et al.* [20] further enhanced the robustness of network to unknown distortions by replacing a fixed set of distortions by a generator. Zhang *et al.* [37] used generative adversarial network (GAN) to optimize the perceptual quality of steganographic images. These methods are usually with good hiding security, i.e., the secret information is unlikely to be detected by steganalysis tools, however, they can only hide a small amount of data.

Image hiding is an important research direction of steganography, which attempts to hide a whole image into another one. Different from the aforementioned methods, it usually requires large hiding capacity. Baluja [4, 5] firstly proposed to hide a whole color image within another one using deep neural networks. To achieve this, a preparation network is developed to extract useful features of the secret image, and then a hiding network is used to fuse the features of secret image within the cover image. Finally, a revealing network is adopted to recover the original secret image. Based on [4], Rahim *et al.* [23] added a regular loss to ensure joint end-to-end training. However, both of them have the problem of color distortion. Zhang *et al.* [38] mitigated this impact by decreasing the payload of secret images, i.e., only embedding gray-scale images. Weng *et al.* [32] further applied this technology to video steganography by temporal residual modeling. The previous works demonstrate that deep networks have great potential in image hiding.

2.2. Invertible Neural Network

Invertible neural network (INN) was first proposed by Dinh *et al.* [9]. Given a variable $\boldsymbol{y}$ and the forward computation $\boldsymbol{x} = f_{\boldsymbol{\theta}}(\boldsymbol{y})$, one can recover $\boldsymbol{y}$ directly by $\boldsymbol{y} = f_{\boldsymbol{\theta}}^{-1}(\boldsymbol{x})$, where the inverse function $f_{\boldsymbol{\theta}}^{-1}$ is designed to share same parameters $\boldsymbol{\theta}$ with $f_{\boldsymbol{\theta}}$. To make INN better handle image-related tasks, Dinh *et al.* [10] introduced convolutional layers in coupling models, and multi-scale layers to reduce the computational cost and increase the regularization ability. Kingma *et al.* [15] introduced invertible 1×1 convolution to INN and proposed Glow, which is efficient on realistic-looking synthesis and manipulation of images.

Due to the excellent performance, INN has been utilized in many image-related tasks. Specifically, Ouderaa *et al.* [28] applied INN to image-to-image translation task. Ardizzone *et al.* [3] introduced conditional INN to guided image generation and colorization, in which the inverse process was guided by a conditional parameter. Xiao *et al.* [33] attempted to find a mapping between low and high resolution images using INN for image rescaling. Lugmayr *et al.* [18] proposed a normalizing flow-based method via INN on super-resolution, which attempted to directly account for the ill-posed nature of super-resolution, and learn to predict diverse photo-realistic high-resolution images. Most recently, Wang *et al.* [30] applied INN in digital image compression task. However, to the best of our knowledge, there is no work to

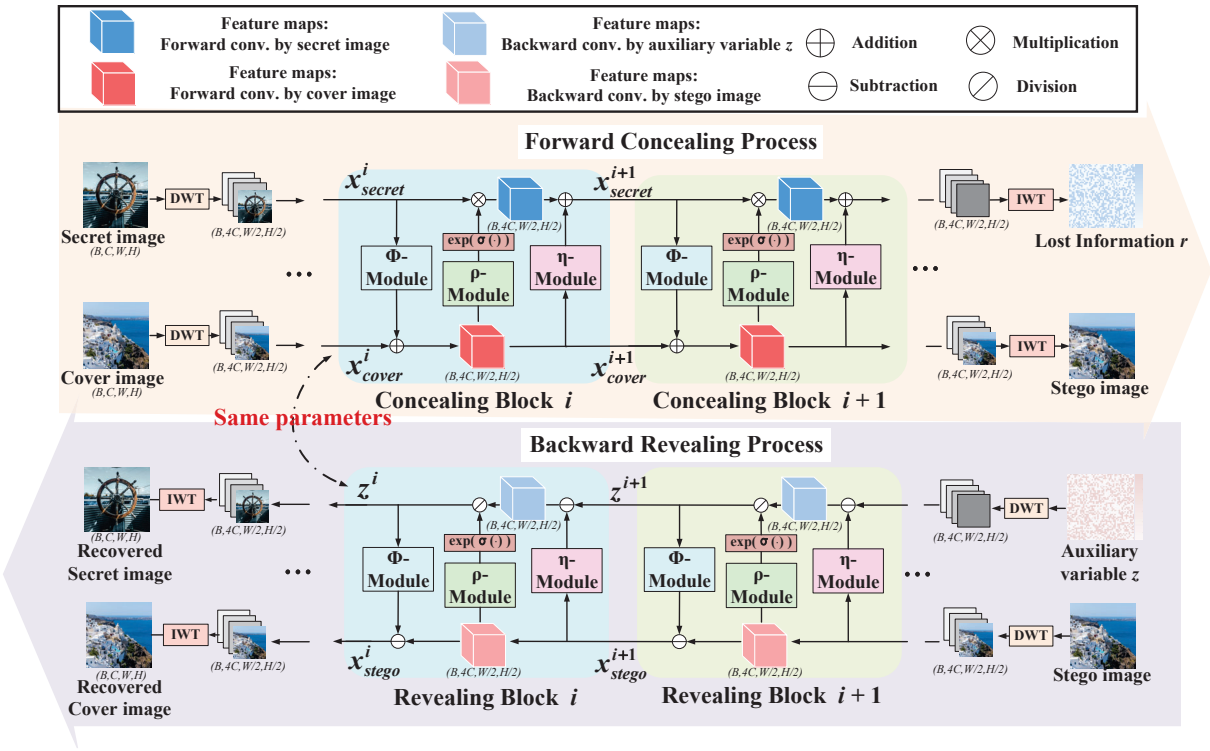

Figure 2. The framework of HiNet. In the forward concealing process, a secret image is hidden in a cover image through several concealing blocks to generate a stego image, together with the lost information. In the backward revealing process, the stego image and an auxiliary variable $\boldsymbol{z}$ from Gaussian distribution are fed to a series of revealing blocks to recover the secret image. Note that in our HiNet, revealing is the inverse process of concealing, and thus they share the same network parameters.

explore INN in image hiding task.

3. Methods

In this section, we propose a novel invertible concealing-revealing network called HiNet to achieve image hiding with large capacity, high security and high invisibility. Table 1 presents the notations used in this paper.

Table 1. Summary of notations in this paper

Notation	Description
$\boldsymbol{x}_{\text{secret}}$	secret image: the image to be hidden
$\boldsymbol{x}_{\text{cover}}$	cover image: the image to hide secret information
$\boldsymbol{x}_{\text{stego}}$	stego image: the image with secret information inside
$\boldsymbol{x}_{\text{rec}}$	recovery image: the recovered secret image from stego image
$\boldsymbol{r}$	lost information: the information lost in concealing process
$\boldsymbol{z}$	auxiliary variable: the variable to help recover stego image

3.1. Network architecture

Fig. 2 shows the overall framework of the proposed HiNet. In the forward concealing process, a pair of secret image $\boldsymbol{x}_{\text{secret}}$ and cover image $\boldsymbol{x}_{\text{cover}}$ are accepted as inputs. They are first decomposed into low and high-frequency wavelet sub-bands through discrete wavelet transform (DWT), which are then fed into a sequence of concealing blocks. The outputs of the last concealing block go through an inverse wavelet transform (IWT) block to generate a stego image $\boldsymbol{x}_{\text{stego}}$, together with the lost information $\boldsymbol{r}$. In the backward revealing process, the stego image $\boldsymbol{x}_{\text{stego}}$ and an auxiliary variable $\boldsymbol{z}$ go through the DWT and a series of revealing blocks to recover the secret image $\boldsymbol{x}_{\text{secret}}$.

Wavelet domain hiding. Image hiding in pixel domain can easily lead to texture-copying artifacts and color distortion [11, 32]. Compared to pixel domain, the frequency domain, especially high-frequency domain, is more appropriate for image hiding. In this paper, we adopt DWT to split image into low and high-frequency wavelet sub-bands before entering the invertible blocks, so that the network can better fuse the secret information into the cover image. Moreover, the perfect reconstruction property of wavelets [21] can help decrease the information loss and improve the image hiding performance. After DWT, the feature map of size (B, C, H, W) is turned into $(B, 4C, H/2, W/2)$, in which B is batch size, H is height, W is width and C is chan-

nel number. As we can see, the computational cost can be reduced by DWT, which can help accelerate the training process. Here, we adopt *Haar* wavelet kernel to perform DWT and IWT, for its simplicity and effectiveness. Note that wavelet transform is bidirectional symmetric, which means it will not affect the end-to-end training of our network.

Invertible concealing and revealing blocks. As shown in Fig. 2, the concealing and revealing blocks have the same sub-modules and share the same network parameters, but with reverse information flow directions. There are M concealing blocks with the same architecture, which is constructed as follows. For the i-th concealing block in the forward process, the inputs are $\boldsymbol{x}^i_{\text{cover}}$ and $\boldsymbol{x}^i_{\text{secret}}$, and the outputs $\boldsymbol{x}^{i+1}_{\text{cover}}$ and $\boldsymbol{x}^{i+1}_{\text{secret}}$ are formulated as follows,

$$\begin{aligned} \boldsymbol{x}^{i+1}_{\text{cover}} &= \boldsymbol{x}^{i}_{\text{cover}} + \boldsymbol{\phi}\left(\boldsymbol{x}^{i}_{\text{secret}}\right) \\ \boldsymbol{x}^{i+1}_{\text{secret}} &= \boldsymbol{x}^{i}_{\text{secret}} \odot \exp\left(\alpha(\boldsymbol{\rho}\left(\boldsymbol{x}^{i+1}_{\text{cover}}\right))\right) + \boldsymbol{\eta}\left(\boldsymbol{x}^{i+1}_{\text{cover}}\right), \end{aligned} \tag{1}$$

where α is a sigmoid function multiplied by a constant factor served as a clamp, and $\odot$ indicates the dot product operation. Here, $\boldsymbol{\rho}(\cdot)$, $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\eta}(\cdot)$ are arbitrary functions and we adopt the widely used dense block in [29] to represent them for its good representation ability. The influence of different architectures for $\boldsymbol{\rho}(\cdot)$, $\boldsymbol{\phi}(\cdot)$, and $\boldsymbol{\eta}(\cdot)$ is discussed in ablation study in Section 4.4. After the last concealing block, we can obtain the outputs $\boldsymbol{x}^{M+1}_{\text{cover}}$ and $\boldsymbol{x}^{M+1}_{\text{secret}}$, which are then fed into two IWT blocks to generate the stego image $\boldsymbol{x}_{\text{stego}}$ and lost information $\boldsymbol{r}$, respectively.

In the revealing process, the information flow direction is from the $(i+1)$-th revealing block to the i-th revealing block, which is in reverse order to the concealing process, as shown in Fig. 2. Specifically, for the M-th revealing block, the inputs are $\boldsymbol{x}^{M+1}_{\text{stego}}$ and $\boldsymbol{z}^{M+1}$ which are generated by the stego image $\boldsymbol{x}_{\text{stego}}$ and an auxiliary variable $\boldsymbol{z}$ through DWT. Here, $\boldsymbol{z}$ is randomly sampled from a Gaussian distribution. The outputs of the M-th revealing block are $\boldsymbol{x}^{M}_{\text{stego}}$ and $\boldsymbol{z}^{M}$. For the i-th revealing block, the inputs are $\boldsymbol{x}^{i+1}_{\text{stego}}$ and $\boldsymbol{z}^{i+1}$, and the outputs are $\boldsymbol{x}^{i}_{\text{stego}}$ and $\boldsymbol{z}^{i}$. Their relationship is modelled as follows,

$$\begin{aligned} \boldsymbol{z}^{i} &= \left(\boldsymbol{z}^{i+1} - \boldsymbol{\eta}\left(\boldsymbol{x}^{i+1}_{\text{stego}}\right)\right) \odot \exp\left(-\alpha(\boldsymbol{\rho}\left(\boldsymbol{x}^{i+1}_{\text{stego}}\right))\right) \\ \boldsymbol{x}^{i}_{\text{stego}} &= \boldsymbol{x}^{i+1}_{\text{stego}} - \boldsymbol{\phi}\left(\boldsymbol{z}^{i}\right). \end{aligned} \tag{2}$$

After the last revealing block, i.e., the revealing block 1, the output $\boldsymbol{x}^{1}_{\text{stego}}$ is fed into an IWT block to generate the recovery image $\boldsymbol{x}_{\text{rec}}$.

The lost information r and auxiliary variable z. The lost information $\boldsymbol{r}$ is one of the outputs in the forward concealing process, and $\boldsymbol{z}$ is one input to the backward revealing process. In the concealing process, the network tries to hide the secret image into the cover image. However, it is difficult to hide such a large capacity in the cover image, which inevitably leads to the loss of secret information. In addition, the intrusion of secret image may destroy the original information in the cover image. The lost secret information and destroyed cover information make up the lost information $\boldsymbol{r}$. Here, $\boldsymbol{r}$ is assumed to be case-agnostic for the reasons below. Suppose that the distribution of all images in dataset is $\mathcal{X}$. The inputs in the forward process are $\boldsymbol{x}_{\text{cover}}$ and $\boldsymbol{x}_{\text{secret}}$, which are sampled from the same dataset and thus follow the same distribution: $\boldsymbol{x}_{\text{cover}}, \boldsymbol{x}_{\text{secret}} \sim \mathcal{X}$. Due to the strict equivalence of Eqs. (1) and (2), and the reversible constraint of INN, the mixed distribution of the outputs $\boldsymbol{x}_{\text{stego}}$ and $\boldsymbol{r}$ should obey the same distribution as inputs, i.e., $\boldsymbol{x}_{\text{stego}} \times \boldsymbol{r} \sim \mathcal{X}$. For stego image $\boldsymbol{x}_{\text{stego}}$, the concealing loss in Section 3.2 pushes its distribution to match the cover image, i.e., $\boldsymbol{x}_{\text{stego}} \sim \mathcal{X}$. Thus, it is reasonable to assume the remained $\boldsymbol{r}$ to be case-agnostic.

In backward revealing, the recovery image $\boldsymbol{x}_{\text{rec}}$ is required to be extracted from only the stego image $\boldsymbol{x}_{\text{stego}}$ with no access to $\boldsymbol{r}$. This is actually an ill-posed problem, because there can be millions of $\boldsymbol{x}_{\text{rec}}$ recovered from the same $\boldsymbol{x}_{\text{stego}}$. In order to obtain the accurate $\boldsymbol{x}_{\text{rec}}$, an auxiliary variable $\boldsymbol{z}$ is adopted in the backward revealing process. The variable $\boldsymbol{z}$ is randomly sampled from a case-agnostic distribution, which is supposed to obey the same distribution as $\boldsymbol{r}$. The distribution is learned during training through the revealing loss in Section 3.2, ensuring that every sample in the distribution is able to well recover the secret information. Here, without loss of generality, we assume the distribution as Gaussian distribution, i.e., $\boldsymbol{z} \sim N\left(\mu_0, \sigma_0^2\right)$.

Why INN works for image hiding? The image hiding task is composed of two reverse procedures: the concealing procedure aims to hide a secret image $\boldsymbol{x}_{\text{secret}}$ in a cover image $\boldsymbol{x}_{\text{cover}}$, to generate a new container called stego image $\boldsymbol{x}_{\text{stego}}$; while the revealing procedure attempts to recover the secret image from the stego image as high-fidelity as possible. In previous works [5, 23, 32], the concealing and revealing procedures are sequentially achieved by two forward networks, i.e., one network for concealing and the other for revealing. However, for perfect concealing and revealing performance, these two processes should be fully reversible. Based on this, we innovatively treat the image concealing and revealing as the forward and backward processes of the same INN, i.e, they are invertible. As a result, they can coordinate with each other to improve the hiding and revealing performance simultaneously. As demonstrated in the experiments, our network with INN architecture significantly advances the state-of-the-art image hiding performance.

3.2. Loss function

The total loss function is composed of three different losses: the concealing loss to guarantee the concealing performance, the revealing losses to ensure the recovering performance, and a new low-frequency wavelet loss to enhance the hiding security.

Concealing loss. The forward concealing process aims

to hide $\boldsymbol{x}_{\text{secret}}$ into $\boldsymbol{x}_{\text{cover}}$, to generate a stego image $\boldsymbol{x}_{\text{stego}}$. The stego image is required to be indistinguishable from the cover image. Toward this goal, the concealing loss L_{con} is defined as follows,

$$L_{\text{con}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ell_{\mathcal{C}} \left(\boldsymbol{x}_{\text{cover}}^{(n)}, \boldsymbol{x}_{\text{stego}}^{(n)} \right), \tag{3}$$

where $\boldsymbol{x}_{\text{stego}}^{(n)}$ is equal to $f_{\boldsymbol{\theta}} \left((\boldsymbol{x}_{\text{cover}}^{(n)}, \boldsymbol{x}_{\text{secret}}^{(n)}) \right)$, with $\boldsymbol{\theta}$ indicating the network parameters. In addition, N is the number of training samples, and $\ell_{\mathcal{C}}$ measures the difference between cover and stego images, which can be ℓ_1 or ℓ_2 norm.

Revealing loss. In the backward revealing process, given the stego image $\boldsymbol{x}_{\text{stego}}$ generated from the forward concealing process, the network should be able to recover the secret image using any sample of $\boldsymbol{z}$ from the Gaussian distribution $p(\boldsymbol{z})$. To achieve this goal, we define the revealing loss L_{rev} as follows,

$$L_{\text{rev}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})} [\ell_{\mathcal{R}} \left(\boldsymbol{x}_{\text{secret}}^{(n)}, \boldsymbol{x}_{\text{rec}}^{(n)} \right)], \tag{4}$$

where the recovery image $\boldsymbol{x}_{\text{rec}}^{(n)}$ is equal to $f_{\boldsymbol{\theta}}^{-1} \left(\boldsymbol{x}_{\text{stego}}^{(n)}, \boldsymbol{z} \right)$, with $f_{\boldsymbol{\theta}}^{-1}(\cdot)$ indicating the backward revealing process. Similar to $\ell_{\mathcal{C}}$, $\ell_{\mathcal{R}}$ measures the difference between recovered secret images $\boldsymbol{x}_{\text{rec}}$ and ground-truth secret images $\boldsymbol{x}_{\text{secret}}$.

Low-frequency wavelet loss. In addition to the above two losses, we propose a low-frequency wavelet loss L_{freq} to enhance the network's anti-steganalysis ability. The motivation of this loss is from [4], which verifies that the information hidden in high-frequency components is less likely to be detected than that in low-frequency components. Here, in order to ensure most information is hidden in the high-frequency sub-bands, the low frequency sub-bands of stego image are required to be similar to those of cover image after wavelet decomposition. Suppose that $\mathcal{H}(\cdot)_{LL}$ indicates the operation of extracting low-frequency sub-bands after wavelet decomposition, the low-frequency wavelet loss L_{freq} is defined as follows,

$$L_{\text{freq}}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ell_{\mathcal{F}} \left(\mathcal{H}(\boldsymbol{x}_{\text{cover}}^{(n)})_{LL}, \mathcal{H}(\boldsymbol{x}_{\text{stego}}^{(n)})_{LL} \right). \tag{5}$$

Here, $\ell_{\mathcal{F}}$ measures the difference between the low-frequency sub-bands of cover and stego images.

Total loss function. The total loss function L_{total} is a weighted sum of concealing loss L_{con}, revealing loss L_{rev} and low-frequency wavelet loss L_{freq}, as follows,

$$L_{\text{total}} = \lambda_c L_{\text{con}} + \lambda_r L_{\text{rev}} + \lambda_f L_{\text{freq}}. \tag{6}$$

Here, λ_c, λ_r and λ_f are weights for balancing different loss terms. In the training process, we firstly pre-train the network by minimizing L_{con} and L_{rev}, i.e., λ_f is set to 0. Then, we add L_{freq} to train the network in an end-to-end manner.

4. Experiments

4.1. Experimental Settings

Datasets and settings. The DIV2K [1] training dataset is used for training our HiNet. The testing datasets include DIV2K [1] testing dataset with 100 images at resolution 1024×1024, ImageNet [25] with 50,000 images at resolution 256×256, and COCO [17] dataset with 5,000 images at resolution 256×256. Note that the testing images are cropped using center-cropping strategy, to make sure the cover and secret images are with the same resolution. The number of concealing and revealing blocks $\boldsymbol{M}$ is set to 16. The training patch size is 256×256, and the number of total iteration is $80K$. The parameters λ_c, λ_r and λ_f are set to 10.0, 1.0, 10.0, respectively. The mini-batch size is set to 16, in which half is randomly selected as cover patches and the remained are secret patches. The Adam [14] optimizer is adopted with standard parameters and an initial learning rate of $1 \times 10^{-4.5}$, which is halved every $10K$ iterations.

Benchmarks. To verify the effectiveness of our method, we compare it with several state-of-the-art (SOTA) image hiding methods, including one traditional image steganography method named 4bit-LSB, and three deep learning based methods: HiDDeN [41], Weng *et al.* [32], and Baluja [5]. For fair comparison, we re-trained the models of Weng *et al.* [32], Baluja [5], and HiDDeN [41] using the same training dataset as ours. Note that the original HiDDeN [41] model can only hide messages, which is not consistent with the image hiding configuration in this paper. To make it able to hide images, we slightly modified its output dimension and then re-trained the network.

Evaluation metrics. There are four metrics adopted to measure the quality of cover/stego and secret/recovery pairs, including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [31], Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). The larger value of PSNR, SSIM and smaller value of RMSE, MAE indicate higher image quality. In addition, we use the statistical steganalysis tool named StegExpose [7] and SRNet [8] to evaluate the security performance of our method.

4.2. Comparison against SOTA methods

Quantitative results. Table 2 compares the numerical results of our HiNet with 4bit-LSB, HiDDeN [41], Weng *et al.* [32] and Baluja [5]. As can be seen from Table 2, our HiNet significantly outperforms other methods in terms of all the four metrics for both cover/stego and secret/recovery pairs. Specifically, for cover/stego image pairs, our HiNet achieves 9.24 dB, 7.63 dB and 6.98 dB improvement in PSNR than the second best results on DIV2K, COCO and ImageNet datasets, respectively. For secret/recovery image pairs, we provide 13.93 dB, 8.29 dB and 10.30 dB PSNR improvement than the second bests on DIV2K, COCO and

Table 2. Benchmark comparisons on different datasets, with the best results in red and second bests in blue.

Methods	Cover/Stego image pair											
	DIV2K				COCO				ImageNet			
	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓
4bit-LSB	33.19	0.9453	6.90	7.95	33.79	0.9479	7.31	9.12	33.68	0.9401	6.46	8.48
HiDDeN [41]	35.21	0.9691	6.98	6.82	36.71	0.9876	6.58	8.73	34.79	0.9380	6.12	7.33
Weng *et al.* [32]	39.75	0.9765	3.24	4.85	38.89	0.9762	3.99	5.91	37.62	0.9588	4.70	5.25
Baluja [5]	36.77	0.9645	3.79	5.02	36.38	0.9563	5.98	7.43	36.59	0.9520	5.61	5.41
HiNet	48.99	0.9971	1.33	1.94	46.52	0.9961	1.87	2.92	44.60	0.9928	2.52	3.62

Algorithms	Secret/Recovery image pair											
	DIV2K				COCO				ImageNet			
	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓
4bit-LSB	30.81	0.9020	8.96	8.01	32.04	0.9127	7.61	9.59	31.26	0.9033	7.71	9.76
HiDDeN [41]	36.43	0.9696	6.02	5.50	37.68	0.9845	4.72	6.33	35.70	0.9601	4.57	6.92
Weng *et al.* [32]	38.93	0.9683	3.95	5.16	38.69	0.9756	4.06	5.95	36.48	0.9537	4.98	6.28
Baluja [5]	35.88	0.9377	4.68	6.11	35.01	0.9341	6.52	8.00	34.13	0.9247	5.31	8.37
HiNet	52.86	0.9992	0.56	0.86	46.98	0.9957	1.60	2.66	46.78	0.9952	1.94	2.74

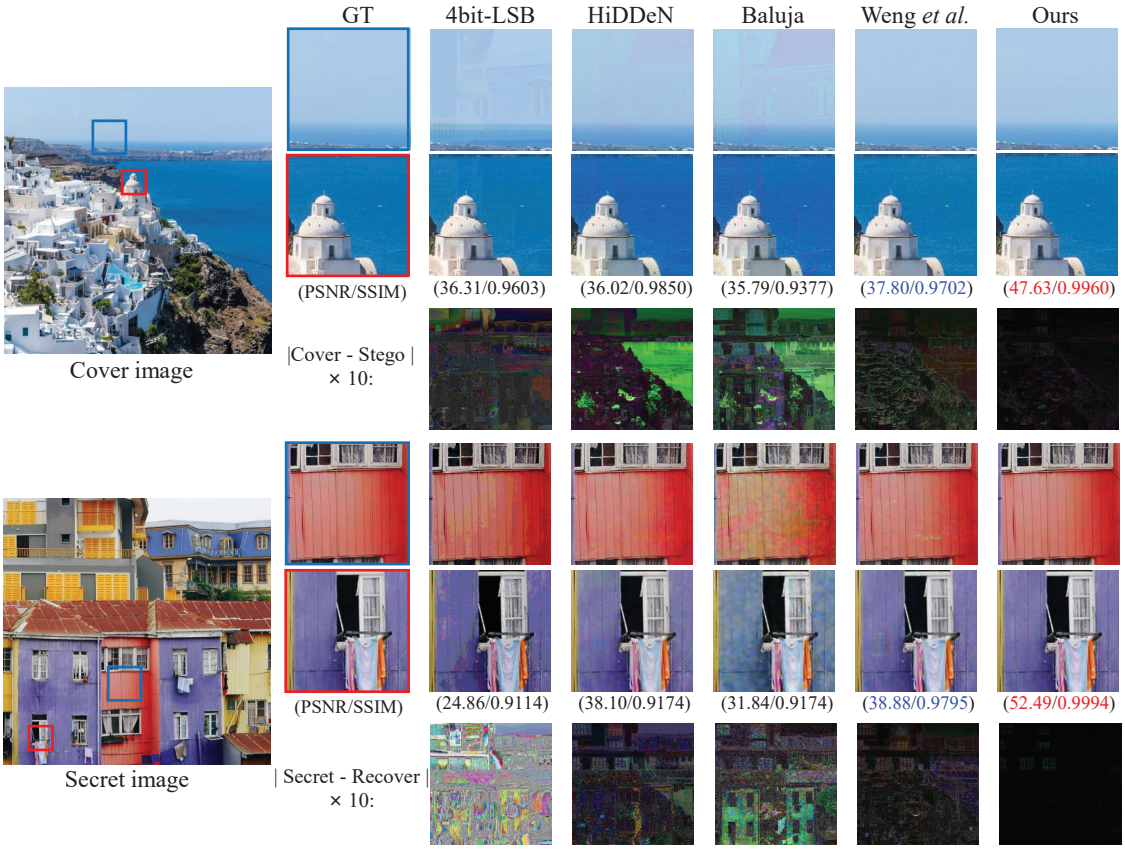

Figure 3. Visual comparisons of stego and recovery images of our HiNet and the comparison methods 4bit-LSB, HiDDeN [41], Baluja [5], and Weng *et al.* [32]. The upper three rows show the enlarged stego images, while the lower three rows show the enlarged recovery images of different methods.

ImageNet datasets, respectively. In addition to PSNR, similar improvements can be seen in SSIM, RMSE and MAE. We achieve significantly better results than the SOTA deep learning based methods, thanks to the reversibility of our HiNet architecture and the wavelet transform which can greatly improve the hiding performance.

Qualitative Results. Fig. 3 compares the stego and recovery images of our HiNet and other four methods. As can be seen, in our method, the difference between cover and stego images is nearly invisible, indicating that we are able to successfully conceal the secret image in the cover image. In addition, our method can nearly perfectly recover the secret

image, i.e., the residual map between the recovery image and the ground-truth secret image is nearly all in black. In contrast, the stego images of 4bit-LSB, HiDDeN [41] and Baluja [5] have obvious texture-copying artifacts, especially in smooth regions. In addition, their recovery images often contain undesirable color deviation problem, in which Weng *et al.* [32] also shows visible blurring artifacts. Compared to these methods, our HiNet not only offers high recovery accuracy, but also enjoys high color fidelity without text-copying artifacts both in the stego and recoveries.

Generalization ability. Although our model is trained only using DIV2K dataset, it offers excellent results on COCO and ImageNet datasets, as shown in Table 2. This demonstrates the good generalization of our model, which is of significant importance in practical applicaitons.

4.3. Steganographic analysis

Steganographic analysis measures the security of stego images, which is an important evaluation part in image hiding task. Specifically, steganalysis measures the possibility to distinguish stego image from cover image by steganalysis tools [22]. The mainstream steganalysis methods can be divided into two categories: traditional statistical methods and new deep learning based methods.

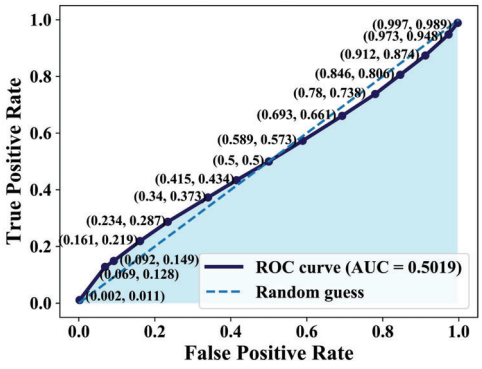

Figure 4. The ROC curve produced by StegExpose for our HiNet.

Statistical steganalysis. We follow [5] to use an open-source steganalysis tool, called StegExpose [7], to measure our model's anti-steganalysis ability. Specifically, we randomly select 2,000 cover and secret images from the testing set and generate the stego images using our HiNet. Then, the secret images are recovered from stego images by our HiNet. To draw the receiver operating characteristic (ROC) curve, we vary the detection thresholds in a wide range in StegExpose [7]. Fig. 4 shows the ROC curve of our HiNet. We can see that the value of area under ROC curve (AUC) is 0.5019, indicating that the detection accuracy is quite close to the random guess. This demonstrates that the stego images generated by our model are with high security, which are able to fool the StegExpose tool with high probability.

Table 3. The detection accuracy using SRNet

Methods	Accuracy (%) ±std
4bit-LSB	99.96 ± 0.06
Baluja [5]	99.67 ± 0.01
HiDDeN [41]	76.49 ± 0.11
Weng *et al.* [32]	75.03 ± 0.59
HiNet	**55.86** ± 0.27

Deep learning based steganalysis. SRNet [8] is a network for image steganalysis, to distinguish stego image from cover image. Table 3 presents the detection accuracy using SRNet for different image hiding methods. Here, the closer the detection accuracy is to 50% (random guess), the better the image hiding algorithm performs. As can be seen, our HiNet achieves 55.86 % detection accuracy, which is significantly better than other SOTA methods [5, 32, 41]. Since 55.86 % is quite close to 50%, the stego image of our method is nearly in-detectable from the cover image.

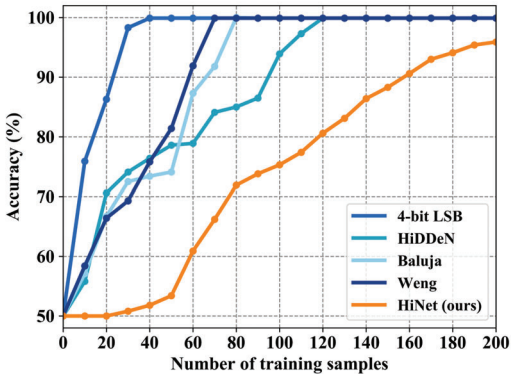

Figure 5. Investigation on the anti-steganalysis ability of different methods. Note that the closer the accuracy is to 50%, the higher anti-steganalysis ability it can achieve.

In addition to the aforementioned steganalysis method, Weng *et al.* [32] proposed a new way for image steganalysis. Specifically, the SRNet is re-trained with different number of cover/stego image pairs generated by one specific model, to investigate how many training images are needed to make SRNet capable to detect stego images. Following [32], we gradually increase the amount of training images to re-train the SRNet. Fig. 5 shows the change of detection accuracy with the number of training images. As we can see from this figure, our HiNet achieve much lower detection accuracy compared to other methods, which indicates the higher anti-steganalysis ability of our method.

4.4. Ablation Study

Effectiveness of wavelet transform. As shown in Table 4, the wavelet transform plays an important role in improving the performance of our method. Specifically, as we can

Table 4. Effectiveness of wavelet transform and low-frequency wavelet loss. The third row represents our HiNet.

Wavelet transform	L_{freq} loss	Cover/Stego image pair				Secret/Recovery pair				Detection rate (%)
		PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	PSNR(dB)↑	SSIM↑	MAE↓	RMSE↓	
✗	✗	40.71	0.9789	3.72	5.16	44.22	0.9938	1.93	3.22	74.25
✓	✗	44.23	0.9918	2.51	3.68	**48.52**	**0.9973**	**1.32**	**2.19**	75.42
✓	✓	**46.52**	**0.9961**	**1.87**	**2.92**	46.98	0.9957	1.60	2.66	**55.86**

see from the first and second rows in Table 4, the PSNR value with wavelet transform increases by 3.52 dB and 4.30 dB for cover/stego and secret image/recovery pairs, respectively. The possible reason is that wavelet transform can successfully separate the low-frequency and high-frequency sub-bands, making it more effective for information hiding.

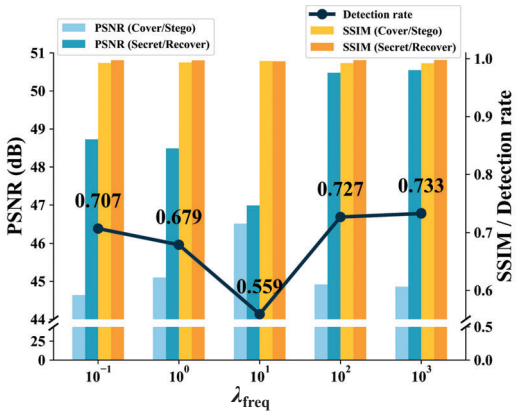

Figure 6. Ablation study on L_{freq} loss. The models are trained with different λ_{freq} values with the other parameters fixed.

Effectiveness of L_{freq} loss. The L_{freq} loss is designed to guarantee that most of secret information is hidden in the high-frequency sub-bands of the cover image, so that the stego image can be less detectable. As demonstrated in the second and third rows in Table 4, L_{freq} significantly improves the security of our method, i.e., the detection rate is decreased from 75.42% to 55.86%. Moreover, with L_{freq}, the average PSNR value of cover/stego image pair is increased by 2.29 dB. Fig. 6 shows the influence of different λ_{freq} on the performance of our method. We can see that when $\lambda_{\text{freq}} = 10^1$, the best trade-off between PSNR, SSIM and detection rate can be obtained.

Influence of $\rho(\cdot)$, $\phi(\cdot)$ and $\eta(\cdot)$ architecture. Fig. 7 shows the effects of different architectures of $\rho(\cdot)$, $\phi(\cdot)$ and $\eta(\cdot)$ on the performance of our method using violin plots. The violin plots visualise the distribution of PSNR value and its probability density (blue area). The top, middle and bottom lines represent maximum, average and minimum PSNR values, respectively. Here, we adopt three typical block architectures, including convolutional, residual and dense blocks, to analyze the influence of different structures on the performance. For fair comparison, these three blocks are constructed to contain same number of parameters. As we can see from this figure, dense block produces the best PSNR results, which is also the reason why we adopt it for $\rho(\cdot)$, $\phi(\cdot)$ and $\eta(\cdot)$ in this paper.

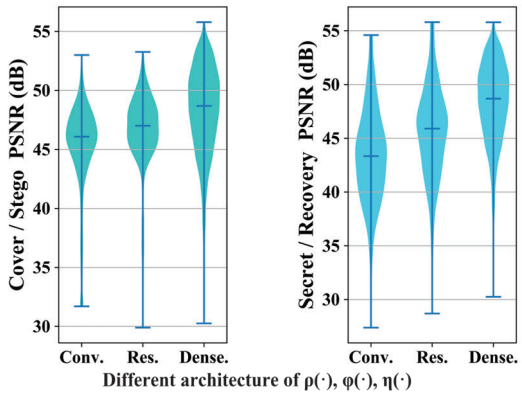

Figure 7. The influence of different architectures of $\rho(\cdot)$, $\phi(\cdot)$ and $\eta(\cdot)$ visualised by violin plots.

5. Conclusion

In this paper, we propose a novel invertible neural network named HiNet for image hiding, which drastically increases both the hiding security and recovering accuracy. Our HiNet models the image concealing and revealing as the forward and backward processes of an invertible network, which means that they share the same network parameters. As a consequence, the network only needs to be trained once to get all network parameters for both image concealing and revealing processes. In network training, a new low-frequency wavelet loss is proposed to improve the security of image hiding. Extensive experimental results show that our method can achieve image hiding with large capacity and high security, which significantly outperforms other SOTA methods both quantitatively and qualitatively.

Acknowledgments. This work was sponsored by CAAI-Huawei Mindspore Open Fund, NSFC under Grants 62050175, 62001016, 61876013, and 61922009, and Beijing Natural Science Foundation under Grant JQ20020.

References

[1] Eirikur Agustsson and Radu 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*, pages 126–135, 2017. 5

[2] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020. 1, 2

[3] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019. 2

[4] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, pages 2069–2079, 2017. 1, 2, 5

[5] Shumeet Baluja. Hiding images within images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 02 2019. 1, 2, 4, 5, 6, 7

[6] Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Transactions on Image Processing*, 10(5):783–791, 2001. 1, 2

[7] Benedikt Boehm. Stegexpose - A tool for detecting LSB steganography. *CoRR*, abs/1410.6656, 2014. 5, 7

[8] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018. 5, 7

[9] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[11] Jessica Fridrich, Miroslav Goljan, and Rui Du. Detecting lsb steganography in color, and gray-scale images. *IEEE Multimedia*, 8(4):22–28, 2001. 1, 2, 3

[12] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*, pages 1954–1963, 2017. 2

[13] Chiou-Ting Hsu and Ja-Ling Wu. Hidden digital watermarks in images. *IEEE Transactions on Image Processing*, 8(1):58–68, 1999. 1, 2

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[15] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 2

[16] Daniel Lerch-Hostalot and David Megías. Unsupervised steganalysis based on artificial training sets. *Engineering Applications of Artificial Intelligence*, 50:45–59, 2016. 1, 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 5

[18] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In *European Conference on Computer Vision*, pages 715–732. Springer, 2020. 2

[19] Weiqi Luo, Fangjun Huang, and Jiwu Huang. Edge adaptive image steganography based on lsb matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2):201–214, 2010. 1, 2

[20] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13548–13557, 2020. 1, 2

[21] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. 3

[22] Ruohan Meng, Qi Cui, and Chengsheng Yuan. A survey of image information hiding algorithms based on deep learning. *Computer Modeling in Engineering & Sciences*, 117(3):425–454, 2018. 7

[23] Rafia Rahim, M Shahroz Nadeem, et al. End-to-end trained cnn encode-decoder networks for image steganography. *arXiv preprint arXiv:1711.07201*, 2017. 1, 2, 4

[24] JJKO Ruanaidh, WJ Dowling, and Francis M Boland. Phase watermarking of digital images. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 239–242. IEEE, 1996. 1, 2

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[26] Abdelfatah A Tamimi, Ayman M Abdalla, and Omaima Al-Allaf. Hiding an image inside another image using variable-rate steganography. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(10), 2013. 2

[27] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2126, 2020. 2

[28] Tycho FA van der Ouderaa and Daniel E Worrall. Reversible gans for memory-efficient image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2019. 2

[29] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 4

[30] Yaolong Wang, Mingqing Xiao, Chang Liu, Shuxin Zheng, and Tie-Yan Liu. Modeling lost information in lossy image compression. *arXiv preprint arXiv:2006.11999*, 2020. 2

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to

structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[32] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 87–95, 2019. 1, 2, 3, 4, 5, 6, 7

[33] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[34] Jianhua Yang, Kai Liu, Xiangui Kang, Edward K Wong, and Yun-Qing Shi. Spatial image steganography based on generative adversarial network. *arXiv preprint arXiv:1804.07939*, 2018. 2

[35] Yang Yang. Basn–learning steganography with binary attention mechanism. *arXiv preprint arXiv:1907.04362*, 2019. 1, 2

[36] Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont. Steganography using a 3 player game. *arXiv preprint arXiv:1907.06956*, 2019. 1, 2

[37] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019. 2

[38] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. *Multimedia Tools and Applications*, 78(7):8559–8575, 2019. 2

[39] Zhuo Zhang, Guangyuan Fu, Fuqiang Di, Changlong Li, and Jia Liu. Generative reversible data hiding by image-to-image translation via gans. *Security and Communication Networks*, 2019, 2019. 1, 2

[40] Ziqiang Zheng, Hongzhi Liu, Zhibin Yu, Haiyong Zheng, Yang Wu, Yang Yang, and Jianbo Shi. Encryptgan: Image steganography with domain transform. *arXiv preprint arXiv:1905.11582*, 2019. 1, 2

[41] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–672, 2018. 1, 2, 5, 6, 7