PixelSteganalysis: Pixel-wise Hidden Information Removal with Low Visual Degradation

Dahuin Jung, Ho Bae, Hyun-Soo Choi, and Sungroh Yoon*, *Senior Member, IEEE*

Abstract—Recently, the field of steganography has experienced rapid developments based on deep learning (DL). DL based steganography distributes secret information over all the available bits of the cover image, thereby posing difficulties in using conventional steganalysis methods to detect, extract or remove hidden secret images. However, our proposed framework is the first to effectively disable covert communications and transactions that use DL based steganography. We propose a DL based steganalysis technique that effectively removes secret images by restoring the distribution of the original images. We formulate a problem and address it by exploiting sophisticated pixel distributions and an edge distribution of images by using a deep neural network. Based on the given information, we remove the hidden secret information at the pixel level. We evaluate our technique by comparing it with conventional steganalysis methods using three public benchmarks. As the decoding method of DL based steganography is approximate (lossy) and is different from the decoding method of conventional steganography, we also introduce a new quantitative metric called the destruction rate (DT). The experimental results demonstrate performance improvements of 10–20% in both the decoded rate and the DT.

Index Terms—Image steganalysis, Active steganalysis, Active warden, Pixel distribution, Image steganography

1 Introduction

Steganography is the technique of unnoticeably concealing a secret message within a plain cover image to covertly send a message to an intended recipient [1]. When a secret message is hidden in a cover image, the output is called a stego image. With the upsurge of big data on the Internet, the threat of unauthorized and unlimited information transaction and display has risen sharply. Furthermore, steganography has been used by international terrorist organizations, various companies, and military organizations for covert communications [2]. It is known that some terrorist groups use steganography to exchange secret messages [3]; it is also being used to steal confidential information from companies [4].

In the process of covertly embedding a secret message into a cover image, the original cover image is marginally altered to become a stego image [1], [5]. In conventional steganography, the payload of the secret message is small, and secret messages are mostly embedded in the least significant bits (LSBs) of the cover image in order to avoid statistical and visual detection [6], [7]. Hence, the secret messages hidden using conventional steganography could be removed via relatively simple steganalysis techniques such as JPEG compression [8]. The decoding method of conventional steganography is lossless, therefore it can well retain the content of the hidden text.

As deep learning (DL) techniques have demonstrated great performance in various fields, so do steganography techniques exploiting DL techniques [9], [10]. The currently proposed DL based steganography disperses the representations of secret images across all the available bits [11] and is not restricted to LSBs. The payload of a secret message embedded using a DL based steganography method is comparatively large; however, because the decoding method of DL based steganography is approximate (lossy), secret messages are mostly limited to the image form.

Steganalysis is the detection, extraction, or destruction of a secret message hidden in a stego image [12], [13]. Depending upon the privilege levels, steganalysis can be categorized as passive or active [14]. Passive steganalysis algorithms aim to determine whether an image contains a secret message. Most passive steganalysis algorithms look for features associated with a particular steganography technique (i.e., non-blind technique). However, active steganalysis algorithms involve blind techniques having the privilege to modify images. Active steganalysis represents the techniques used to extract and/or remove secret messages. However, most active steganalysis approaches try to remove the secret messages because extracting the exact messages is difficult in general cases [15]. The original images processed after using active steganalysis must be nearly as unchanged as possible because not all images are stego images. That is, a good active steganalysis technique should aim to remove a secret message as much as possible while introducing minimal changes to the appearance of the image. To that end, we propose a new method of removing the secret image by restoring the distribution of the stego image to that of the original cover image. In Fig. 1, we illustrate an overview of steganography and active steganalysis with symbols of each process and material. The contributions of this paper are as follows:

- To the best of our knowledge, this is the first method that effectively removes secret hidden images using a DL based steganography method. This is the first study utilizing DL to restore stego images to the original cover images.

- *D. Jung, and S. Yoon are with the Dept. of Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea.*
- *H. Bae is with Dept. of Cyber Security, Ewha Womans University, Seoul, 03760, Korea.*
- *H.S. Choi is with Dept. of Computer Science & Engineering and Interdisciplinary Graduate Program in Medical Bigdata Convergence in Kangwon National University, Chuncheon, 24341, Korea, and also with Ziovision, Chuncheon, 24341, Korea.*
- *S. Yoon is also with Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul 08826, Korea*
- *Correspondence should be addressed to S. Yoon. (sryoon@snu.ac.kr).*

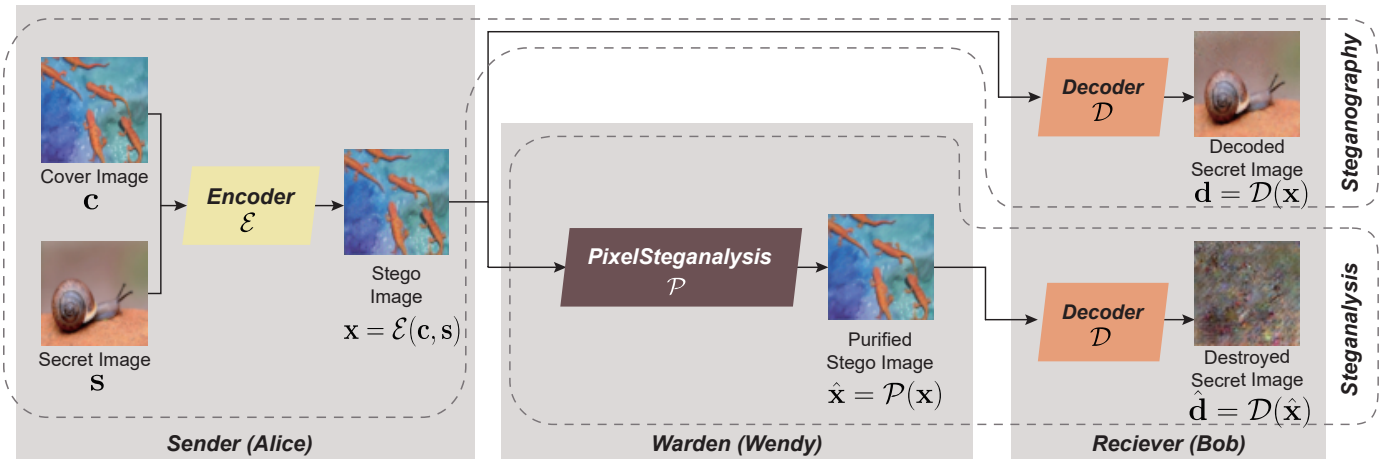

Fig. 1: How steganography works (Encoder $\mathcal{E}$ and Decoder $\mathcal{D}$) and how active steganalysis can disturb it. The destroyed secret image $\hat{\mathbf{d}}$ shows that PixelSteganalysis $\mathcal{P}$ disrupts a covert transmission between a sender and a receiver with an imperceptible difference on the stego image.

We present a theoretical formulation of the problem and its objectives. Furthermore, we experimentally show the possibility of using our approach as a passive steganalysis technique.
- We assume a real-world situation in which access to either the cover image or the secret image is not allowed. Our framework only utilizes a dataset commonly associated with the target society (neither the cover image nor the secret image itself).
- We propose a new evaluation metric called the destruction rate (DT) suitable for evaluating the performance of active steganalysis against the DL based steganography methods with lossy characteristics.
- Our method outperforms conventional active steganalysis from both DL based steganography and conventional steganography, with both high and low payloads. Compared with the adaptive Gaussian noise method, our method exhibited improvements of up to 18% and 20% in terms of the peak signal-to-noise ratio (PSNR) and DT, respectively.

The remainder of this paper is organized as follows. Section 2 provides a brief description of related and comparison works. Section 3 provides detailed descriptions of the attack scenario, problem formulation, and proposed methodology. Section 4 suggests a new evaluation metric, DT, that complements the limitation of the conventional evaluation metric. Section 5 demonstrates the proposed methodology through various combinations of experiments. Finally, Section 6 discusses the results and areas of future study.

2 Background

2.1 Conventional Active Steganalysis

The destruction of the secret message hidden using conventional steganography was straightforward. Thus, active steganalysis has not been developed after several simple yet effective steganalysis approaches were proposed. The most basic approach is to take N LSB planes of the stego image and flip the bits [16]. Another commonly used strategy for destructing a secret message is to overwrite the LSB bits randomly using Gaussian noise or other noise [8]. Although applying adaptive randomization into LSB bits can have comparably high destruction capability on spatial steganography algorithms, yet, at the same time, it can largely harm the image quality. Furthermore, filter-based constructive destruction approaches have been proposed [14], [17]–[19]. First, denoising is used to remove the hidden secret messages, considering the secret message as a noise added to the cover image [17]. In addition, Wiener restoration is a representative method for conventional active steganalysis [14]. The median filter (a denoising technique) and Wiener restoration both operate quite optimally on frequency domain steganography algorithms. We compared adaptive randomization, denoising, and Wiener restoration methods with the steganalysis method proposed in this paper because these three methods have been demonstrated as effective on conventional steganography [19].

2.2 Conventional and DL based Steganography

Steganography, unlike watermarking [20], aims at covert transmission through invisible data hiding. For it, various methods have been proposed to increase both the hiding capability and invisibility. However, the hiding capability and invisibility have a contradictory relation in the field of steganography [21]. Because conventional steganography is aimed at perfect invisibility, an extremely small hiding capability is generally maintained. The LSB insertion [1], [22] is the most conventional steganographic algorithm. However, it is statistically obvious. Recent studies have proposed more advanced approaches that design diverse distortion functions to maintain image statistics. The Highly Undetectable steGO (HUGO) method is the first steganography method that devises a distortion function [6], [23]. The HUGO method embeds the secret information at the positions where the difference between the features of the cover and stego images in the SPAM feature space is low, such as at an edge. In the spatial domain, the Wavelet Obtained Weights (WOW) method computes the sums of the weighted changes in the horizontal, vertical, and diagonal wavelet coefficients. The WOW method, then, estimates a pixel whose probability of being revealed is high in at least one direction by using a reciprocal Holder norm [7]. On the basis of the estimated pixels, the WOW method can avoid clean edge areas while embedding secret information. As a disadvantage of WOW, the embedding cost is not sufficiently sensitive to texture regions

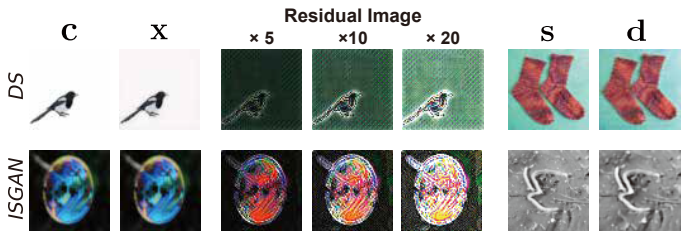

Fig. 2: Samples of DS [11] and ISGAN [9]. The labels x5, x10, and x20 represent the magnification ratio (five, ten, and 20 times, respectively) of the residual image. The first row shows the residual outputs between the cover image $\mathbf{c}$ and the stego image $\mathbf{x}$ generated by DS. We can observe that the diagonal grid pattern is distributed all over the background of the residual images. The second row shows the residual outputs between the cover image $\mathbf{c}$ and the stego image $\mathbf{x}$ generated by ISGAN. For ISGAN, the stego image $\mathbf{x}$ looks natural despite the large difference between the cover image $\mathbf{c}$ and stego image $\mathbf{x}$. However, the illumination of the decoded secret image $\mathbf{d}$ is noticeably deviated from the secret image $\mathbf{s}$.

because of merely adding the reciprocal norm rather than relative changes to the wavelet coefficient [24]. Furthermore, the Spatial - UNIversal WAvelet Relative Distortion (S-UNIWARD) method is similar to the WOW method. However, the S-UNIWARD method addressed and moderated the above mentioned disadvantages of the WOW method [25].

Several DL based steganography approaches attempted to hide secret information with a very small payload [26]–[28]. However, as DL developed, DL based steganography began to embed secret messages of bigger payloads, such as a full-size image, into the cover image, with improved capacity [9]–[11], [29]–[40]. DL based steganography focuses on hiding a much larger amount of secret information in the cover image by relaxing the constraint of perfect invisibility. StegNet [10] proposed an additional loss term named variance loss, which can reduce the noisiness of a stego image produced by generator networks. ISS-GAN [39] introduced a cycle discriminative structure and the concept of inconsistent loss, both of which can improve the quality and security of a stego image. Deep Steganography (DS) [11] involves the additional use of a prep-network to transform a color-based secret image to an edge-based secret image for more natural and compact embedding. ISGAN [9] uses only the Y component from the YCbCr color space of the cover image to hide secret gray images, so that the destruction of the hidden secret images is more difficult than that using other methods.

The DL based steganography approaches cannot be detected easily using conventional passive steganalysis because these approaches tend to maintain the pixel distributions of the original image to the greatest extent possible [9], [26], which was demonstrated experimentally by Baluja and Shumeet [11]. We also test whether conventional and DL based passive steganalysis methods can detect stego images created by DL based steganography methods. For testing, we use two representative statistical passive steganalysis techniques, RS [41] and SPA [42], and one DL based passive steganalysis technique, YeNet [43]. We confirm experimentally that RS, SPA, and YeNet were unable to determine the stego images created by DL based steganography methods when assuming real-world situations in which no access to the utilized steganography algorithm is allowed.

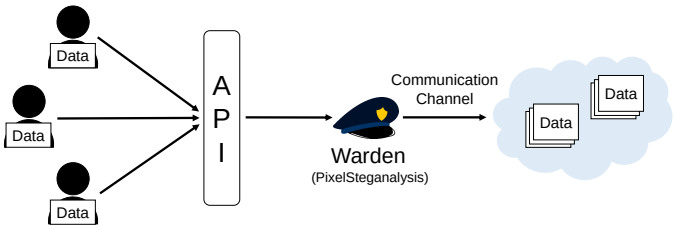

Fig. 3: The attack scenario of the covert transmission of restricted information, based on Simmons' Prisoners' Problem [45].

Yu and Chong [39] demonstrated that secret images hidden using DL based steganography cannot be removed using conventional active steganalysis. Similar to conventional steganography methods, DL based steganography methods also hide the majority of secret information in the high-frequency areas of the cover image, such as edge, where the bandwidth is sufficiently wide to hide a considerable amount of information naturally. It is also known from information theory that a change in the high-frequency area is hard to discover [44]. More specifically, in the case of DS, 48.32% of the secret image is hidden in the edge areas on average, notwithstanding the fact that the edge area is only a small part of the entire image (approximately only 20% in a natural image). Moreover, in the case of ISGAN, 34.47% of the secret image is hidden in the edge areas on average. UDH [40] recently proposed a meta-architecture that can disentangle the encoding of the secret image from the cover image. They conducted further analysis about where and how the secret image is encoded. A visual sample in which secret information is heavily hidden in the edge areas is provided in Fig. 2. Based on the analysis results, we propose a method that can remove secret information hidden using DL based steganography as much as possible while reducing the loss of the original cover image. In addition, we experimentally confirm the effectiveness of our approach on conventional steganography.

3 Proposed Method

3.1 Scenario

There is always a risk that individuals having access to sensitive or proprietary information try to leak information and then share it with competitors or adversaries. We particularly assume a scenario that the local hosts in restricted environments such as companies inside try to leak hidden information via the internet. As depicted in Fig. 3, our framework can be located in the company's uplink to the internet outside as an active warden [45].

An active warden has the privilege to alter the content of the communication to confuse hidden data within the carrier. However, its privilege is limited to slight changes [19]. As an active warden, our method aims at removing the hidden secret message in a direction of restoring the distribution of the original images.

3.2 Problem Formulation

We represent the stego image, original cover image, secret image, and decoded secret image as $\mathbf{x}$, $\mathbf{c}$, $\mathbf{s}$, and $\mathbf{d}$, respectively. Then, we can formulate the encoding and decoding algorithms of steganography as $\mathbf{x} = \mathcal{E}(\mathbf{c}, \mathbf{s})$ and $\mathbf{d} = \mathcal{D}(\mathbf{x})$, respectively. We also represent a purified stego image after performing active steganalysis as $\hat{\mathbf{x}}$, and a destroyed secret image after performing active steganalysis as $\hat{\mathbf{d}} = \mathcal{D}(\hat{\mathbf{x}})$. We first formulate the objective

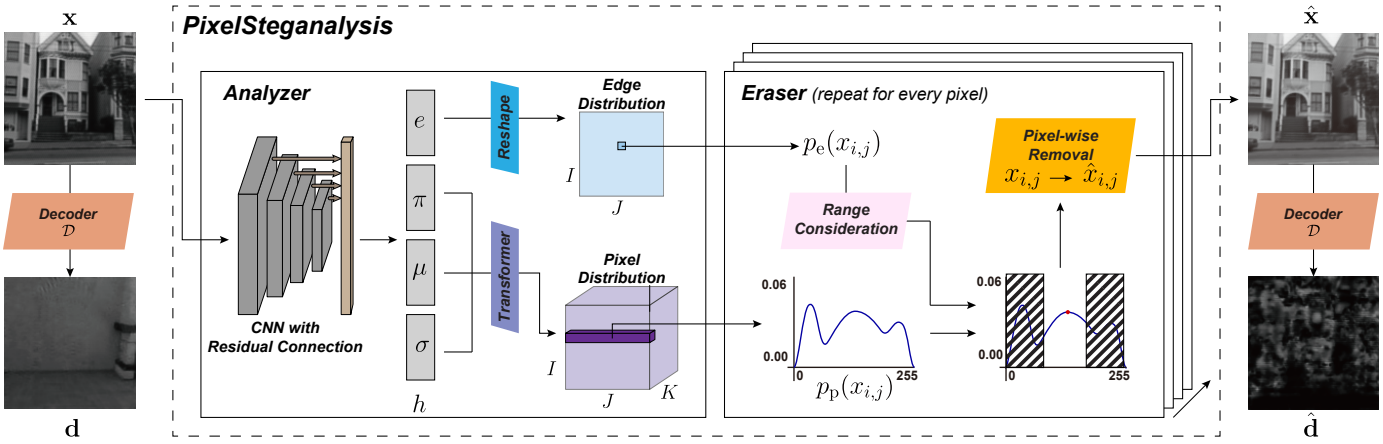

Fig. 4: Model overview. Our framework, PixelSteganalysis $\mathcal{P}$, consists of the *analyzer* and *eraser*. PixelSteganalysis $\mathcal{P}$ receives the stego image $\mathbf{x}$ and produces the purified stego image $\hat{\mathbf{x}}$ ($\hat{\mathbf{x}} = \mathcal{P}(\mathbf{x})$).

of this study by using these representations. From the perspective of active steganalysis, the ultimate goal is to find the purified stego image $\hat{\mathbf{x}}$ as follows:

$$\underset{\hat{\mathbf{x}}}{\text{minimize}} \quad \|\hat{\mathbf{x}} - \mathbf{c}\|_\infty . \tag{1}$$

If $\mathbf{x}$ returns to $\mathbf{c}$, it indicates that the hidden secret image is removed completely after performing active steganalysis. In other words, the hidden secret image is completely extracted from the stego image $\mathbf{x}$. However, since the original cover image $\mathbf{c}$ and secret image $\mathbf{s}$ are inaccessible in the process of detecting or extracting the hidden secret image, it is almost impossible. As described in Sec. 3.1, rather than an utterly open scenario, we assume the scenario that works as an active warden in the restricted environment. Each company can accumulate and make use of the data samples transmitted outside from their uplink to build a dataset $\mathbf{X}$ having a similar distribution to the original cover image $\mathbf{c}$. With these assumptions, we can alter Eq. 1 to

$$\begin{aligned} &\underset{\hat{\mathbf{x}}}{\text{maximize}} \quad p^*(\hat{\mathbf{x}}) \\ &\text{subject to} \quad \|\hat{\mathbf{x}} - \mathbf{x}\|_\infty \leq \epsilon_{\max}, \end{aligned} \tag{2}$$

where $p^*(\cdot)$ denotes the exact distribution of the training dataset $\mathbf{X}$ and $\epsilon_{\max}$ denotes the allowed maximum modification of each pixel. That is, we aim to find an image $\hat{\mathbf{x}}$ that maximizes $p^*(\hat{\mathbf{x}})$ subject to the constraint that $\hat{\mathbf{x}}$ is within the $\epsilon_{\max}$-ball of $\mathbf{x}$. We constrain the maximum pixel modification because most steganography methods aim at a minimum change in $\mathbf{x}$ from $\mathbf{c}$, both statistically and visually. In practice, we propose an adaptive consideration degree of modification, e_{norm}, bounded by $\epsilon_{\max}$ as the constraint per pixel. This adaptive consideration range allows more changes in the edge areas and less change in the non-edge areas. Detailed descriptions are given in Eqs. 5 and 6. We also approximate p^* to a PixelCNN++ distribution p (Eq. 3). To maximally utilize the intrinsic characteristics of steganography, we propose a technique that sequentially satisfies the constrained objective in Eq. 2 at the pixel level, instead of using gradient-based constrained optimization such as L-BFGS-B [46] or image generation [47].

3.3 Proposed Algorithm

To remove a hidden image, our proposed algorithm requires neither the knowledge of the utilized steganography algorithm (blind) nor the distribution of the original cover image, while offering minimum perceptual degradation and even perceptual improvement of the stego image. As illustrated in Fig. 4, our algorithm employs an *analyzer* and an *eraser* to produce the purified stego image, $\hat{\mathbf{x}}$. The *analyzer* takes the stego image as input and produces an edge distribution, $\mathbf{p}_{\text{e}}(\mathbf{x})$, and the distribution of all the pixels, $\mathbf{p}_{\text{p}}(\mathbf{x})$, of the given image (Sec. 3.3.1 *Analyzer*). The generated distributions are then used to remove the secret image hidden in the stego image by using the *eraser* (Sec. 3.3.2 *Eraser*). Note that the candidate input images are not limited to grayscale. However, in this section, for easier visualization and explanation, we assume grayscale input images.

3.3.1 Analyzer

The *analyzer* obtains $\mathbf{p}_{\text{p}}(\mathbf{x})$ and $\mathbf{p}_{\text{e}}(\mathbf{x})$ by employing a neural network trained using a dataset $\mathbf{X}$ having similar distribution as the original cover images. The auto-regressive models learn the image distribution. Then, we calculate the marginal likelihood of an image by taking the product of the probabilities of each sampled pixel:

$$p(\mathbf{x}) = \prod_{i=2}^{I \times J} p\left(\mathbf{x}(i) | \mathbf{x}(1:(i-1))\right), \tag{3}$$

where $I \times J$ denotes the height and width of the image. Therefore, we can take advantage of the explicit distribution of all the pixels, unlike other modeling algorithms. Moreover, $\mathbf{p}_{\text{e}}(\mathbf{x})$ is the information indicating the high frequency areas of the image. $\mathbf{p}_{\text{e}}(\mathbf{x})$ is jointly learned and is utilized in the *eraser*. To obtain $\mathbf{p}_{\text{p}}(\mathbf{x})$ and $\mathbf{p}_{\text{e}}(\mathbf{x})$, we propose a CNN architecture inspired by PixelCNN++ [48], which is the most representative DL based auto-regressive model.

As described in Fig. 4, the activation of the last fully connected layer of the *analyzer* is named *h* and consists of trained parameters e, π, μ, and σ. Using the parameters $\pi, \mu,$ and σ trained with a dataset $\mathbf{X}$ that has a distribution similar to the original cover image, we obtain a discretized Gaussian mixture likelihood for all pixels $\mathbf{p}_{\text{p}}(\mathbf{x}|\pi, \mu, \sigma)$ ($\in \mathbb{R}^{I \times J \times K}$) obtained by learning the distribution of the dataset $\mathbf{X}$, $p(\cdot)$, in an auto-regressive way, where K denotes a pixel depth dimension (0–255). This procedure is referred to as the *transformer*. Using the trained $\mathbf{p}_{\text{p}}$, we can obtain how appropriate the current pixel value is, provided the previous pixel values with respect to the distribution of cover

images in the shape of a Gaussian mixture model. The operation of the *transformer* is based on PixelCNN++. The detail of the *transformer* can be found in supplementary S2 and [48].

We also train a network to detect high frequency areas in which one can embed secret information unnoticeably, which we call an edge distribution, $\mathbf{p}_{\mathrm{e}}(\mathbf{x})$. As shown in Fig. 4, $\mathbf{p}_{\mathrm{e}}(\mathbf{x})$, the vector e of h is reshaped into $I \times J$. $\mathbf{p}_{\mathrm{e}}(\mathbf{x})$ is used to determine the consideration range of the suspicious information per pixel in the *eraser*, as explained in Sec. 2.

We minimize the sum of the *image loss*, $\mathcal{L}_I$, and the *edge loss*, $\mathcal{L}_E$, to obtain $\mathbf{p}_{\mathrm{p}}$ and $\mathbf{p}_{\mathrm{e}}$, respectively. $\mathcal{L}_I$ is the negative log-likelihood of the image obtained by the product of the conditional distribution of each pixel, and $\mathcal{L}_E$ is the mean-squared error between the results obtained using a conventional edge detector and those obtained using our neural network. We have

$$\begin{aligned} \mathcal{L}_I &= -\mathbb{E}_{\mathbf{x}\sim\mathbf{X}} \log p(\mathbf{x}), \\ \mathcal{L}_E &= \mathbb{E}_{\mathbf{x}\sim\mathbf{X}}[\frac{1}{I \times J}\sum_{i=1}^{I}\sum_{j=1}^{J}(\mathbf{p}_{\mathrm{d}}(\mathbf{x})(i,j) - \mathbf{p}_{\mathrm{e}}(\mathbf{x})(i,j))^2], \\ \mathcal{L} &= \lambda_I \mathcal{L}_I + \lambda_E \mathcal{L}_E, \end{aligned} \tag{4}$$

where $\mathbf{x}$ denotes the image of the training dataset $\mathbf{x} \in \mathbf{X}$, $\mathbf{p}_{\mathrm{d}}(\mathbf{x})$ the edge distribution obtained using a conventional edge detector [49] currently, and $\mathbf{p}_{\mathrm{e}}(\mathbf{x})$ the learned edge distribution. For empirical risk minimization, we make use of empirical expectations of each loss. Moreover, λ_I and $\lambda_E > 0$ denote the hyperparameters used to balance the strength of both the loss terms.

3.3.2 Eraser

The best scenario from the perspective of steganalysis is that both the cover and stego images are accessible. Then, we can easily restore a stego image, $\mathbf{x}$, to a cover image, $\mathbf{c}$, using Eq. 1. However, this is generally impractical. Thus, instead, we suggest an approach for removing the hidden secret image by adjusting the pixel value of the suspicious regions in which the secret image may be hidden, using the pixel level information. In the *eraser*, we aim to find a purified stego image, $\hat{\mathbf{x}}$, that maximizes $p(\hat{\mathbf{x}})$ under the constraint given in Eq. 2. We iteratively substitute the pixel i's value with the neighboring pixel value of the highest probability based on information from pixels $i-1, i-2, ...$

A large amount of secret information is hidden in the edge areas of the cover image. Therefore, we control the consideration range of each pixel, and the range is decided by two factors: ϵ and $\mathbf{p}_{\mathrm{e}}(\mathbf{x})$. We calculate the adaptive consideration degree of modification per pixel as follows:

$$e_{\mathrm{norm}}(i,j) = \epsilon + \lceil \frac{\mathbf{p}_{\mathrm{e}}(\mathbf{x})(i,j)}{\mathbf{p}_{\mathrm{e}}(\mathbf{x})_{\max}} \times (\epsilon_{\max} - \epsilon) \rceil, \tag{5}$$

where i and j denote the pixel coordinates of the image, $\mathbf{p}_{\mathrm{e}}(\mathbf{x})_{\max}$ the maximum edge value and ϵ a hyperparameter representing an allowed degree of the least modification ($\epsilon > 0$). We make use of a ceiling, $\lceil \cdot \rceil$, to keep e_{norm} as an integer. The hyperparameter ϵ is suggested for fair comparisons with other active steganalysis methods and to guarantee the removal of encoded secret information in non-edge areas. Eq. 5 shows that e_{norm} is bounded by ϵ and $\epsilon_{\max}$ ($\epsilon \leq e_{\mathrm{norm}} \leq \epsilon_{\max}$). Eq. 5 guarantees the lower and upper bounds of the consideration range per pixel. $\epsilon_{\max}$ is $2 \times \epsilon$ in our experiments.

The pixel value of the stego image is not deviated significantly from the corresponding pixel value of the cover image. Therefore we only consider probabilities close to those of the given pixel values. We set up the adaptive consideration range of modification per pixel as:

$$\begin{aligned} r_{\min}(i,j) &= \max(\mathbf{x}(i,j) - e_{\mathrm{norm}}(i,j),\ 0), \\ r_{\max}(i,j) &= \min(\mathbf{x}(i,j) + e_{\mathrm{norm}}(i,j),\ 255). \end{aligned} \tag{6}$$

Eq. 6 determines the range of pixel values considered for modification by centering around $\mathbf{x}(i,j)$.

Each pixel is iteratively replaced with the one having the highest probability value among the allowed neighboring pixel values, as follows:

$$\hat{\mathbf{x}}(i,j) = \underset{k \in [r_{\min}(i,j), r_{\max}(i,j)]}{\operatorname{argmax}} \mathbf{p}_{\mathrm{p}}(\hat{\mathbf{x}})(i,j,k), \tag{7}$$

where initially $\hat{\mathbf{x}} = \mathbf{x}$. Eq. 7 removes the secret message at the pixel level based on $\mathbf{p}_{\mathrm{p}}(\mathbf{x})(i,j)$ bounded by $r_{\min}(i,j)$ and $r_{\max}(i,j)$. For every pixel, the pixel distribution $\mathbf{p}_{\mathrm{p}}(\hat{\mathbf{x}})$ should be re-extracted whenever the previous pixel value is modified. However, re-extracting the pixel distribution for all the pixels requires excessive time to modify a single image. Therefore, to decrease the runtime, we propose an approximation of Eq. 7, in which the pixel distribution $\mathbf{p}_{\mathrm{p}}(\mathbf{x})$ is only extracted before the iteration. The approximation is formed as

$$\hat{\mathbf{x}}(i,j) = \underset{k \in [r_{\min}(i,j), r_{\max}(i,j)]}{\operatorname{argmax}} \mathbf{p}_{\mathrm{p}}(\mathbf{x})(i,j,k), \tag{8}$$

for each pixel. This approximation leads to much faster runtime but slightly decreases the quality of the results. A comparison between the original modification case and the approximated modification case is presented in the Sec. 5.2.

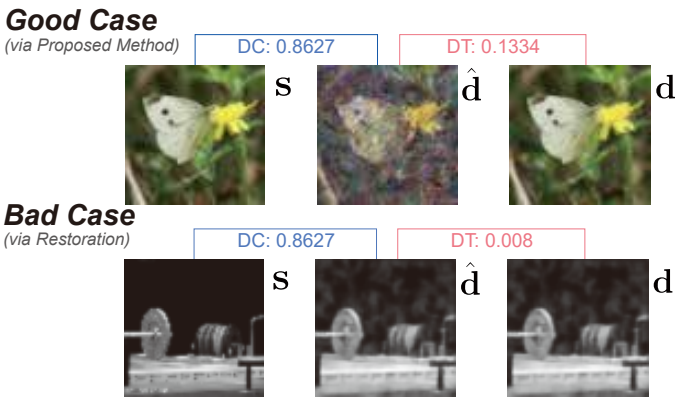

Fig. 5: Top: the success of destruction has a relatively high value of DT. Bottom: the failure of destruction is indicated by the almost zero DT values. However, the DC values of the two cases are exactly the same.

4 Proposed Evaluation Metric

The goal of an active steganalysis is the development of a technique that minimally destroys the cover image while effectively removing hidden steganography. To properly examine the performance of an active steganalysis method, an evaluation metric that defines the criteria to be met against various types of steganography approaches is necessary. However, the evaluation method against DL based steganography cannot separate the destruction of the hidden secret image from the degradation of the decoded secret image itself. In detail, the evaluation method against DL

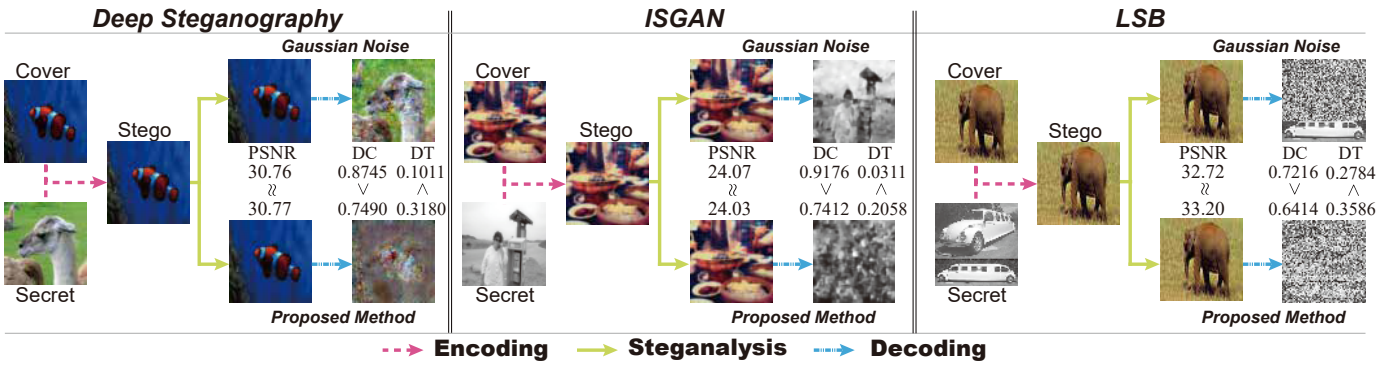

Fig. 6: Three examples of how our method and Gaussian noise differ in efficiency at the same PSNR.

based steganography, the decoded rate (DC) [10], [14], is defined by

$$\text{Decoded Rate} = 1 - \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\left|\mathbf{s}(i,j) - \hat{\mathbf{d}}(i,j)\right|}{I \times J}. \quad (9)$$

A condition that was guaranteed in conventional steganography is $\mathbf{s} = \mathbf{d}$. Therefore, we could use the DC to measure the exact performance of active steganalysis. However, the decoding method of the DL based steganography algorithms is $\mathbf{s} \neq \mathbf{d}$ (lossy). The DC between $\mathbf{s}$ and $\mathbf{d}$ is approximately 90% [9], [11]. In other words, the error rate between $\mathbf{s}$ and $\mathbf{d}$ is already 10%, which depends upon how well the decoding algorithm is trained. Thus, we propose a new evaluation metric called the DT, which can accurately assess the destruction performance of active steganalysis on both conventional steganography and DL based steganography, regardless of the decoding algorithm. The DT is defined as

$$\text{Destruction Rate} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\left|\mathbf{d}(i,j) - \hat{\mathbf{d}}(i,j)\right|}{I \times J}. \quad (10)$$

To produce results independently of the performance of the decoding algorithm, the base image is changed into $\mathbf{d}$ instead of $\mathbf{s}$. DT is a more reliable metric than DC for representing the pure degree of hidden image destruction for each active steganalysis method. For example, as depicted in Fig. 5, it is possible that the DT values can be significantly different, whereas the DC values are exactly the same. Because the DC value is affected by the performance of both active steganalysis and the decoding method of steganography, its value can be meaningful even if the performance of the active steganalysis is poor.

5 Experiments

5.1 Experimental Results

We compare our method with three commonly used conventional steganalysis techniques: Gaussian noise, Denoising, and Restoration. We use the PSNR [50] and the structural similarity (SSIM) [51] to measure the quality of the purified image. PSNR and SSIM are basic metrics for comparing the quality of the purified image to that of the original cover image. The SSIM results are provided in supplementary S5.

Among the DL based steganography algorithms, we made use of two representative methods, DS, ISGAN and UDH, to compare our method with the existing active steganalysis techniques. Additionally, we test our proposed steganalysis method on non-DL steganography techniques, namely, LSB insertion, HUGO, WOW, and S-UNIWARD.

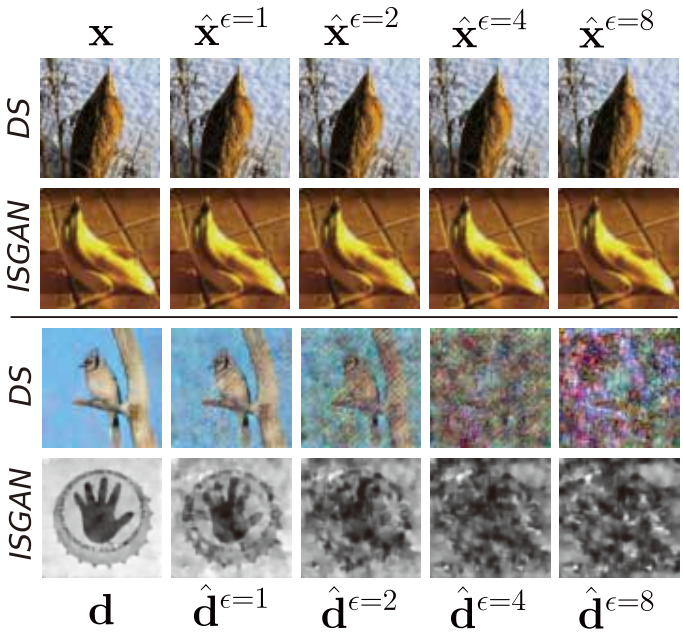

Fig. 7: According to the increase of ϵ, the degree of destruction of the secret image hidden in the stego image (ImageNet) generated by DS and ISGAN after applying our method.

We made use of three datasets: Cifar-10, Boss1.0.1, and ImageNet. The results for the three datasets show similar trends. The details of the settings and the comparison methods are described in supplementary S1 and S6.

As shown in Fig. 6, our method removes the hidden secret images better than the Gaussian noise method at the same PSNR in the three cases of steganography (DS, ISGAN, and LSB insertion). As we can observe, the DT value demonstrates the performance of active steganalysis more accurately. We also plot quantitative results for all the three datasets in Fig. 8. In Fig. 8, the PSNR at $\epsilon = 0$ represents the quality of the unpurified stego image compared with the cover image. Unlike the Gaussian noise method, the PSNR of our method did not fall below a certain level, even at $\epsilon = 8$ for all datasets. The reason is that we closely follow the distribution of the original cover image, so that regardless of how large ϵ is, the distribution of the purified stego image would not be substantially different from that of the cover image.

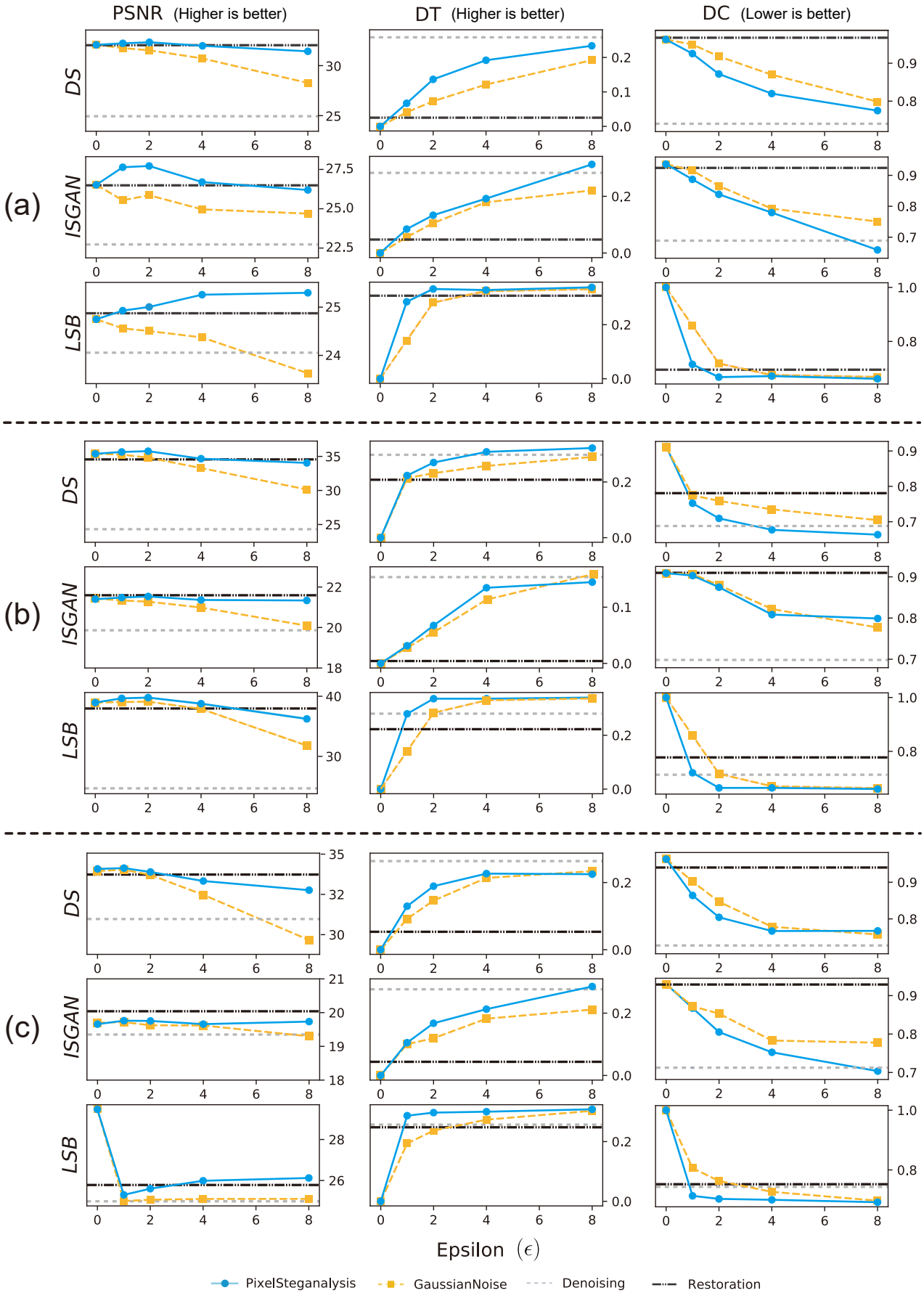

Fig. 8: Experimental results of our work and other steganalysis methods on the (a) Cifar-10, (b) ImageNet, and (c) Boss1.0.1 datasets. The higher the PSNR is, the better the preservation of the original cover image is. The lower the DC is and the higher the DT is, the better the destruction of the hidden secret image is.

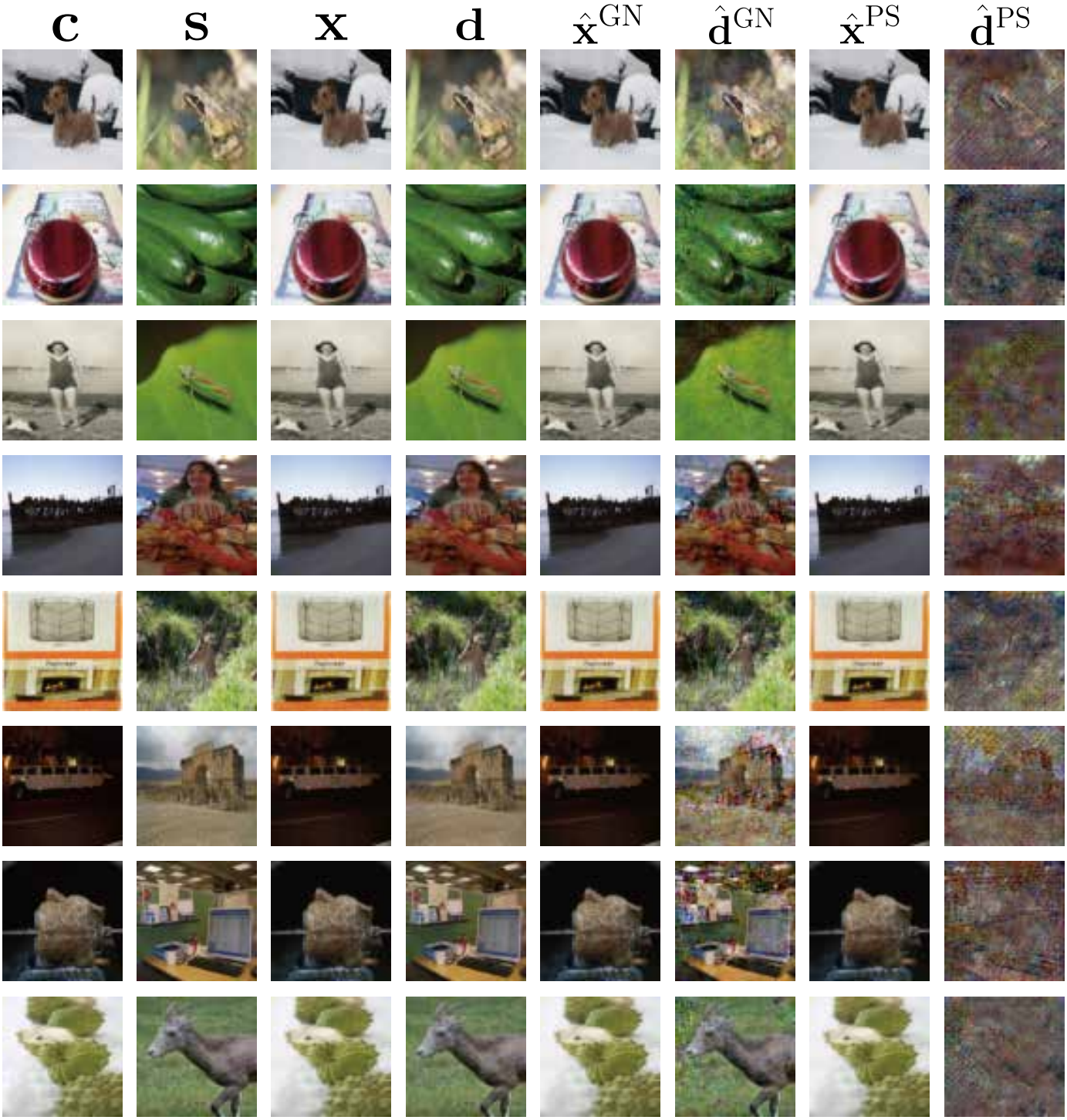

Fig. 9: Comparison of the destructed degree of the secret image decoded by the stego image from our method with that of the secret image decoded by the stego image from Gaussian noise **(DS on ImageNet)**. $\hat{\mathbf{x}}^{\mathrm{GN}}$ represents stego image modified by Gaussian noise, $\hat{\mathbf{d}}^{\mathrm{GN}}$ represents secret image decoded from $\hat{\mathbf{x}}^{\mathrm{GN}}$, $\hat{\mathbf{x}}^{\mathrm{PS}}$ represents stego image modified by our method, and $\hat{\mathbf{d}}^{\mathrm{PS}}$ represents secret image decoded from $\hat{\mathbf{x}}^{\mathrm{PS}}$.

TABLE 1: PSNR (higher is better) and DC (lower is better) against HUGO(H), WOW(W), and S-UNIWARD(S)

	IMAGENET (PSNR [50])						IMAGENET (DC [14])					
ϵ	2			4			2			4		
non-DL	*H*	*W*	*S*	*H*	*W*	*S*	*H*	*W*	*S*	*H*	*W*	*S*
Proposed	**44.7**	**44.7**	**44.7**	**40.3**	**40.3**	**40.3**	**15.7**	**15.7**	**15.7**	**14.0**	**13.9**	**13.9**
Gaussian Noise	43.7	43.7	43.7	37.8	37.8	37.8	25.0	25.0	25.0	12.5	12.5	12.5
Denoising	H: 26.2, W: 26.2, S: 26.2						H: 28.0, W: 35.3, S: 35.3					
Restoration	H: 45.3, W: 45.3, S: 45.4						H: 22.0, W: 22.0, S: 22.0					

TABLE 2: PSNR of non-stego (benign) images

	IMAGENET				BOSS1.0.1			
ϵ	1	2	4	8	1	2	4	8
Proposed	50.07	**46.30**	**43.26**	**39.15**	55.07	**51.30**	**48.26**	**44.15**
Gaussian Noise	**51.24**	44.13	37.80	31.77	**56.14**	49.13	42.88	36.77
Denoising	25.96				31.28			
Restoration	43.12				47.42			

Our method removes the hidden images better than the other methods, as verified by the DC and DT values (at $\epsilon \leq 8$). Compared with the adaptive Gaussian noise method, our method shows an improvement of up to 18% and 20% in terms of PSNR, and DC and DT, respectively. By analyzing the DT values, we see that for DS and LSB, hidden information is destroyed even for small ϵ values. However, for ISGAN, ϵ should be greater than 2 to see some levels of removal. In addition, we experiment with a conventional steganography technique, LSB insertion. We embedded a full-size gray image into a full-size colored cover image using the LSB insertion technique.

As shown in Fig. 8, we can see that the stego images purified by our method resemble the cover images when increasing the value of ϵ. Also, the degree of removal of the hidden message using our method is similar to or higher than that of the existing steganylsis algorithms. There is no visual degradation of the stego image by applying the proposed method although ϵ is larger, as shown in Fig. 7. Considering the visual degradation and the destruction of the secret images, $\epsilon = 4$ seems a reasonable value for all cases.

In Fig. 9, we compare the visual results when Gaussian noise and PixelSteganalysis are applied against DS as an active warden. Specifically, we compare the degree of destruction of the decoded secret images when the PSNR between the two stego images is almost the same. There is no visual difference between $\hat{x}^{\text{GN}}$ and $\hat{x}^{\text{PS}}$, where GN is Gaussian noise and PS is PixelSteganalysis. However, the results of $\hat{d}$ are completely different. The decoded secret images from $\hat{x}^{\text{GN}}$ are almost preserved, while the decoded secret images from $\hat{x}^{\text{PS}}$ look like noise. As demonstrated in conventional steganography methods [52], [53] and the background section of our manuscript, the encoding and decoding of secret images are dependent on cover images. If only simple texture regions exist in the cover image and the color is monotonous, the encoding of the secret image cannot be more concentrated in texture regions, so a larger ϵ is needed to remove the secret image widely embedded all over the cover image. We can see corresponding examples in the 3rd and 6th rows of Fig. 9.

The conventional active steganalysis algorithms fail at destroying the hidden secret message. A large amount of alteration on stego images degrades the image quality at the same time. Unlike that, our method rather aims at restoring the original cover image from its stego image. Thus, our method can accurately destroy the hidden secret image with less visual degradation. We also test our method against the recently proposed DL based steganography method, UDH. The results are consistent with the results obtained against DS and ISGAN, and it results are given in supplementary S8. More visual samples for Cifar-10 and Boss1.0.1 datasets are shown in supplementary S11.

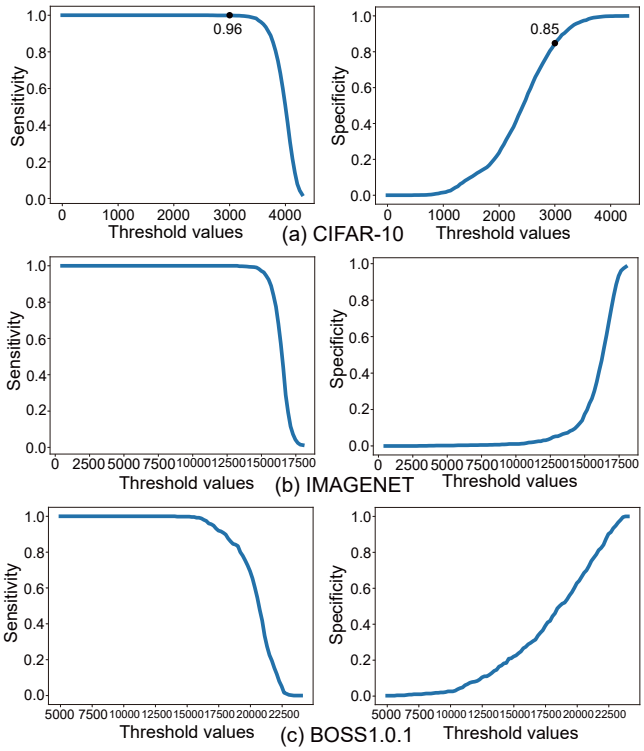

Fig. 10: Sensitivity and specificity of non-stego and stego (DS) images for the (a) Cifar-10, (b) ImageNet, and (c) Boss1.0.1 datasets. To see the possibility of the proposed method as a passive steganalysis technique, we measured the number of modifications on each non-stego and stego image by the proposed method. The number of modifications on stego images is greater than that of non-stego images, especially for the Cifar-10 and Boss1.0.1 datasets.

Analysis about Other Steganalysis Algorithms An effective active steganalysis algorithm should, first, remove as much of a secret message as possible and, second, do so with minimal degradation of the stego image. That is, both conditions must be met. The Gaussian noise and denoising techniques, however, met the first condition but did not satisfy the second one. The restoration method met the second condition but did not satisfy the first one. In the case of Gaussian noise method, if ϵ exceeds 4, the degradation of the stego image is severe, and stego image becomes perceptually noisy, as depicted in Fig. 6 and the supplementary S10. In the case of denoising, the DT is higher than that of our method sometimes, but the PSNR is very low compared to the other steganography methods. In case of restoration, the PSNR is similar to that of our method; however, DT is typically the worst. The visual samples are presented in supplementary S7.

Additional Experiments on Non-DL Steganography We validate our method not only on DL based steganography algorithms but also on more sophisticated conventional non-DL steganography algorithms (e.g., HUGO, WOW, and S-UNIWARD) by using a small payload (1 bpp/ch) on the ImageNet dataset. Our method is comparable to or even better than other

TABLE 3: Ablation results of the proposed methods. For PSNR, SSIM and DT, higher is better. For DC, lower is better.

	ϵ	PSNR [50]	SSIM [51]	DC [14]	DT
Proposed w/o edge	1	35.66	0.9834	0.7761	0.2117
	2	35.72	0.9837	0.7523	0.2365
	4	35.39	0.9811	0.7375	0.2720
Proposed	1	35.89	0.9839	0.7691	0.2184
	2	35.85	0.9842	0.7258	0.2626
	4	35.67	0.9822	0.6923	0.3001

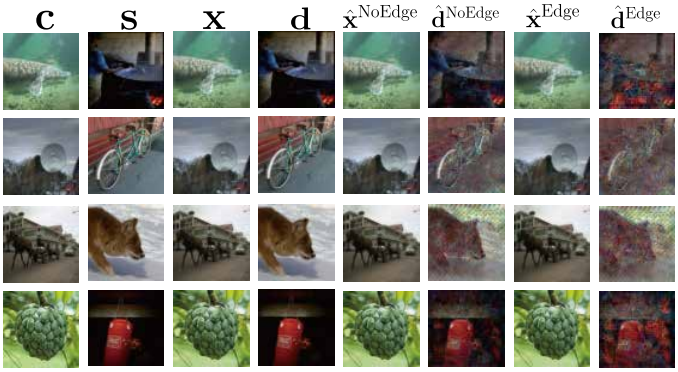

Fig. 11: Ablation study: no edge detection (DS on ImageNet). $\hat{\mathbf{x}}^{\text{NoEdge}}$ represents a stego image modified by PixelSteganalysis but without edge detection, and $\hat{\mathbf{d}}^{\text{NoEdge}}$ represents a secret image decoded from $\hat{\mathbf{x}}^{\text{NoEdge}}$. $\hat{\mathbf{x}}^{\text{Edge}}$ represents a stego image modified by PixelSteganalysis with edge detection, and $\hat{\mathbf{d}}^{\text{Edge}}$ represents a secret image decoded from $\hat{\mathbf{x}}^{\text{Edge}}$.

methods on sophisticated conventional non-DL steganography methods in terms of PSNR and DC, as presented in Tab. 1 (ImageNet) and supplementary S3. Especially, at $\epsilon = 2$, our method outperforms the other existing methods with an amount of DC decreased by 10 or more on the steganography methods, namely, HUGO, WOW, and S-UNIWARD.

Harmless on Non-Stego (Benign) Images In the case of active steganalysis, we assume having the privilege to allow modifications to all the inspected images. We, therefore, experiment on image degradation when applying our method to non-stego (innocuous/benign) images, as summarized in Tab. 2 (ImageNet and Boss1.0.1). After applying our proposed method, there was some degree of image degradation, but the PSNR of our method was always higher than that of the other conventional steganalysis algorithms at $\epsilon \leq 4$. In detail, the standard deviation is less than 1 for all the datasets and ϵ. Our method did not fall below 40 even for the large ϵ as a whole. Degradation of the Cifar-10 dataset after performing active steganalysis is provided in supplementary S4. In contrast to our proposed method, the existing algorithms mostly show visible degradation of benign images.

Potential as Passive Steganalysis If cannot apply our method to every image owing to limited privileges, the method can first operate as a detector (passive steganalysis). Compared to recent advanced passive steganalysis algorithms which assume that access to the cover image, the secret image, or the applied steganography algorithm is required, our algorithm does not need any of them. We measure the total number of modifications of the pixel value on each non-stego and each stego image when applying our method. As a binary classification task, we classify the input image as a stego image when the total number of modifications of the pixel value is greater than or equal to a threshold value, and vice versa. We measure the performance of our method as a detector with various threshold values, as shown in Fig. 10. As the metrics, we use the sensitivity and specificity, which is formulated as follows:

$$\text{Sensitivity} = \frac{\#\text{True Positive}}{\#\text{True Positive} + \#\text{False Negative}}, \tag{11}$$

$$\text{Specificity} = \frac{\#\text{True Negative}}{\#\text{True Negative} + \#\text{False Positive}}, \tag{12}$$

where a True Positive is a stego image that the model correctly predicts as a stego image, a True Negative is a non-stego image that the model correctly predicts as a non-stego image, a False Positive is a non-stego image that the model incorrectly predicts as a stego image, and a False Negative is a stego image that the model incorrectly predicts as a non-stego image.

For Cifar-10, the total number of modifications of the pixel value in the stego images was noticeably higher than that of non-stego images. By setting the threshold value to 3,000, it is possible to obtain a sensitivity and specificity of 0.96 and 0.85, respectively. For ImageNet and Boss1.0.1, the total number of modifications of the pixel value in the stego images does not deviate significantly from the total number of modifications of the pixel value in the non-stego images, as compared to the Cifar-10 case.It is because that ImageNet and Boss1.0.1 consist of more various classes of large images. However, as a stego detector, it is possible to obtain a high sensitivity for all three datasets, as shown in Fig. 10 (a), (b), and (c), with the relatively low threshold values (15,000 for ImageNet and 17,500 for Boss1.0.1). We believe that our method can be used as a primary inspector to find out suspicious images.

5.2 Ablation Studies

No Edge Detection For ablation studies, we test the effectiveness of edge detection. We proceed the experiment using Cifar-10. We can observe that especially at ϵ = 1 and 2, the effect of the edge detection as a guide is large. The visual samples of $\mathbf{p}_{\text{e}}(\mathbf{x})$ from our neural network are provided in supplementary S9. The quantitative and qualitative results are presented in Tab. 3 and Fig. 11. As shown in Tab. 3 and Fig. 11, we could remove more hidden information while keeping the visual degradation of the stego images reduced by making use of edge detection.

No Approximation The results of Fig. 8 are obtained using the approximated version. The original version takes an average of 3 min to purify a single Cifar-10 image. However, the approximated version takes less than 10 ms to process the same image. We compare the DC of both the versions by using stego images generated via DS. The results demonstrate that the average DC of the original version is 75.8% and that of the approximated version is 76.3%. The difference is as small as 0.5%.

6 Conclusions

We propose a DL based steganalysis technique that effectively removes secret images by restoring the distribution of the original image. We use deep neural networks to formulate and solve problems by leveraging sophisticated pixel and edge distributions in the images. In particular, our method shows a remarkable performance against DL based steganography, which is difficult to detect with passive steganalysis in a blind case.

There is a limitation to our method in that it can degrade all of the inspected images. However, when considering an environment

or society in which the confidentiality of sensitive information is extremely important, our method will be a good solution. Compared with existing active steganalysis methods, our approach is the only way to restore a stego image to its original cover image. Therefore, our method is feasible for simultaneously minimizing the degradation of benign images and minimizing false negatives.

In a future study, we will consider replacing $\mathbf{p}_{\mathrm{d}}$ with a metric more suited to the characteristics of a steganography algorithm. We currently use the Prewitt operator as $\mathbf{p}_{\mathrm{d}}$ to locate the high frequency areas of an image. Moreover, instead of using auto-regressive models, it is possible to use other approaches, such as GAN based methods, to model the distribution of the images. However, the distributions of both the original cover images and the stego images are quite similar; therefore, a significant reduction in the PSNR may occur without a careful modification. Thus, the proposed method can carefully remove suspicious traces at the pixel level by maximally utilizing the intrinsic characteristics of steganography.

7 Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [2018R1A2B3001628], the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2021, Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].

References

[1] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, 1998.

[2] M. K. Sharma and P. Gupta, "A comparative study of steganography and watermarking"," *International Journal of Research in IT & Management (IJRIM)*, vol. 2, no. 2, pp. 2231–4334, 2012.

[3] F. Gardner, "How do terrorists communicate?" Nov 2013. [Online]. Available: https://www.bbc.com/news/world-24784756

[4] A. King, "Ge engineer tied to china charged with theft of company secrets," Aug 2018. [Online]. Available: https://asia.nikkei.com/Business/Companies/GE-engineer-tied-to-China-charged-with-theft-of-company-secrets

[5] W.-N. Lie and L. C. Chang, "Data hiding in images with adaptive numbers of least significant bits based on the human visual system," in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 1. IEEE, 1999, pp. 286–290.

[6] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *International Workshop on Information Hiding*. Springer, 2010, pp. 161–177.

[7] V. Holub and J. J. Fridrich, "Designing steganographic distortion using directional filters." in *WIFS*, 2012, pp. 234–239.

[8] J. Fridrich, M. Goljan, and D. Hogea, "Steganalysis of jpeg images: Breaking the f5 algorithm," in *International Workshop on Information Hiding*. Springer, 2002, pp. 310–323.

[9] S. Dong, R. Zhang, and J. Liu, "Invisible steganography via generative adversarial network," *arXiv preprint arXiv:1807.08571*, 2018.

[10] P. Wu, Y. Yang, and X. Li, "Stegnet: Mega image steganography capacity with deep convolutional network," *arXiv preprint arXiv:1806.06357*, 2018.

[11] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Advances in Neural Information Processing Systems*, 2017, pp. 2069–2079.

[12] N. F. Johnson and S. Jajodia, "Steganalysis of images created using current steganography software," in *International Workshop on Information Hiding*. Springer, 1998, pp. 273–289.

[13] M. Bachrach and F. Y. Shih, "Image steganography and steganalysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 3, pp. 251–259, 2011.

[14] P. Amritha, M. Sethumadhavan, and R. Krishnan, "On the removal of steganographic content from images," *Defence Science Journal*, vol. 66, no. 6, pp. 574–581, 2016.

[15] H. Karaman and S. Sagiroglu, "An application based on steganography," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 2012, pp. 839–843.

[16] J. M. Ettinger, "Steganalysis and game equilibria," in *International Workshop on Information Hiding*. Springer, 1998, pp. 319–328.

[17] H. Gou, A. Swaminathan, and M. Wu, "Noise features for image tampering detection and steganalysis," in *2007 IEEE International Conference on Image Processing*, vol. 6. IEEE, 2007, pp. VI–97.

[18] P. L. Shrestha, M. Hempel, T. Ma, D. Peng, and H. Sharif, "A general attack method for steganography removal using pseudo-cfa reinterpolation," in *2011 International Conference for Internet Technology and Secured Transactions*. IEEE, 2011, pp. 454–459.

[19] P. A. Lafferty, *Obfuscation and the steganographic active warden model*. The Catholic University of America, 2008.

[20] I. J. Cox, M. L. Miller, J. A. Bloom, and C. Honsinger, *Digital watermarking*. Springer, 2002, vol. 53.

[21] P. Ramu, R. Swaminathan, *et al.*, "Imperceptibility—robustness tradeoff studies for ecg steganography using continuous ant colony optimization," *Expert Systems with Applications*, vol. 49, pp. 123–135, 2016.

[22] J. Mielikainen, "Lsb matching revisited," *IEEE signal processing letters*, vol. 13, no. 5, pp. 285–287, 2006.

[23] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.

[24] T. Denemark, J. Fridrich, and V. Holub, "Further study on the security of s-uniward," in *Media Watermarking, Security, and Forensics 2014*, vol. 9028. International Society for Optics and Photonics, 2014, p. 902805.

[25] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[26] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Advances in Neural Information Processing Systems*, 2017, pp. 1954–1963.

[27] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "Ssgan: Secure steganography based on generative adversarial networks," in *Pacific Rim Conference on Multimedia*. Springer, 2017, pp. 534–544.

[28] 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.

[29] Z. Wang, N. Gao, X. Wang, J. Xiang, and G. Liu, "Stnet: A style transformation network for deep image steganography," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 3–14.

[30] J. Huang, S. Cheng, S. Lou, and F. Jiang, "Image steganography using texture features and gans," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[31] Z. Wang, N. Gao, X. Wang, J. Xiang, D. Zha, and L. Li, "Hidinggan: High capacity information hiding with generative adversarial network," in *Computer Graphics Forum*, vol. 38, no. 7. Wiley Online Library, 2019, pp. 393–401.

[32] R. Meng, Z. Zhou, Q. Cui, X. Sun, and C. Yuan, "A novel steganography scheme combining coverless information hiding and steganography," *J. Inf. Hiding Privacy Protection*, vol. 1, no. 1, pp. 43–48, 2019.

[33] B. Chen, J. Wang, Y. Chen, Z. Jin, H. J. Shim, and Y.-Q. Shi, "High-capacity robust image steganography via adversarial network." *KSII Transactions on Internet & Information Systems*, vol. 14, no. 1, 2020.

[34] P. Kuppusamy, K. Ramya, S. S. Rani, M. Sivaram, and V. Dhasarathan, "A novel approach based on modified cycle generative adversarial networks for image steganography," *Scalable Computing: Practice and Experience*, vol. 21, no. 1, pp. 63–72, 2020.

[35] A. Das, J. S. Wahi, M. Anand, and Y. Rana, "Multi-image steganography using deep neural networks," *arXiv preprint arXiv:2101.00350*, 2021.

[36] J. Qin, J. Wang, Y. Tan, H. Huang, X. Xiang, and Z. He, "Coverless image steganography based on generative adversarial network," *Mathematics*, vol. 8, no. 9, p. 1394, 2020.

[37] J. Liu, Y. Ke, Z. Zhang, Y. Lei, J. Li, M. Zhang, and X. Yang, "Recent advances of image steganography with generative adversarial networks," *IEEE Access*, vol. 8, pp. 60 575–60 597, 2020.

[38] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster r-cnn," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.

[39] C. Yu, "Integrated steganography and steganalysis with generative adversarial networks," 2018.

[40] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 223–10 234, 2020.
[41] D. Chaum, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *Journal of cryptology*, vol. 1, no. 1, pp. 65–75, 1988.
[42] O. Berthold, H. Federrath, and S. Köpsell, "Web mixes: A system for anonymous and unobservable internet access," in *Designing privacy enhancing technologies*. Springer, 2001, pp. 115–129.
[43] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.
[44] K. Solanki, N. Jacobsen, U. Madhow, B. Manjunath, and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," *IEEE Transactions on image processing*, vol. 13, no. 12, pp. 1627–1639, 2004.
[45] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology*. Springer, 1984, pp. 51–67.
[46] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.
[47] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
[48] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," *arXiv preprint arXiv:1701.05517*, 2017.
[49] J. M. Prewitt, "Object enhancement and extraction," *Picture processing and Psychopictorics*, vol. 10, no. 1, pp. 15–19, 1970.
[50] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.
[51] 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.
[52] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4206–4210.
[53] 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, 2019.