A Novel Gradient-guided Post-processing Method for Adaptive Image Steganography

Guoliang Xie[a,b], Jinchang Ren[b,c,*], Stephen Marshall[a], Huimin Zhao[b], Rui Li[b,d]

[a] *Dept. of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G1 1XQ, U.K.*
[b] *School of Computing Sciences, Guangdong Polytechnic Normal University, Guangzhou, 510640, China*
[c] *National Subsea Center, Robert Gordon University, Aberdeen, AB21 0BH, U.K.*
[d] *School of Art and Design, Guangzhou College of Commerce, Guangzhou, 511363, China*

Abstract

Designing an effective cost function has always been the key in image steganography after the development of the near-optimal encoders. To learn the cost maps automatically, the Generative Adversarial Networks (GAN) are often trained from the given cover images. However, this needs to train two Convolutional Neural Networks (CNN) in theory and is thus very time-consuming. In this paper, without modifying the original stego image and the associated cost function of the steganography, and no need to train a GAN, we proposed a novel post-processing method for adaptive image steganography. The post-processing method aims at the embedding cost, hence it is called Post-cost-optimization in this paper. Given a cover image, its gradient map is learned from a pre-trained CNN, which is further smoothed by a low-pass filter. The elements of the cost map derived from the original steganography are projected to 0,1 for separating embeddable and non-embeddable areas. For embeddable areas, the elements will be further screened by the gradient map, according to the magnitudes of the gradients, to produce a new cost map. Finally, the new cost map is used to generate new stego images. Comprehensive experiments have validated the efficacy of the proposed method, which has outperformed several state-of-the-art approaches, whilst the computational cost is also significantly reduced.

[*] Corresponding author

Keywords:
Neural Networks, Image Steganography, Image Steganalysis, Gradients

1. Introduction

Image steganography is to embed a secret message into a cover image for covert communication, where the sender uses a pre-defined method to embed a secret message, and thus the receiver can extract the message without any faults. There are two types of image steganography, namely, non-adaptive steganography and adaptive steganography [1]. Non-adaptive steganography does not consider the detail of the cover image during embedding, while adaptive steganography usually confines the embedding to more textured areas of the cover image for enhanced security [2]. Nowadays, most researchers work on adaptive steganography for its superior performance in securing secret messages [3, 4, 5, 6, 7].

Recently, Mandal et al. [3] provided a literature survey in digital image steganography, which discussed the challenges and future directions in this area. Muralidharan et al. [4] did a detailed comparison between the development of steganography and steganalysis, which covers more than 150 papers. The battle between steganography and steganalysis can be found easily. For example, Zhang et al. [5] proposed their adaptive robust steganography for open-social-network communication, which tried to implement this technique in everyday communication. To counter this problem, Zhu et al. proposed a deep learning network for steganalysis to destroy the secret messages in their work [6]. More related works can be found in [4].

The adaptive steganography can be roughly divided into two categories, i.e., the model-based and the convolution-based, according to how the cover images are processed. The convolution-based methods are widely used in the design of image steganalysis due to their high efficiency in generating stego images. To name a few, some typical methods include the Wavelet Obtained Weights (WOW) [1], Spatial Universal Wavelet Relative Distortion (S-UNIWARD) [8] and the HIgh-pass, Low-pass, Low-pass (HILL) model [9].

Model-based methods, relying on the statistical correlation among pixels and patches, usually require complicated matrix analysis and hence need a longer processing time [10, 11, 12]. For example, the Highly Undetectable steGO (HUGO) method [13] allocates the embedding information to the textural areas of images by calculating the sum of differences between the

Subtractive Pixel Adjacency Matrix (SPAM) feature vectors [14]. Fridrich et al. [15] adopted a multivariate quantized Gaussian (MG) distribution for determining the cost of embedded pixels by minimizing the Kullback-Leibler (KL) divergence [2] between the cover and the stego images. Qin et al [16] modelled image residuals obtained by high-pass filters with MG to further improve the performance. Recently, inspired by the Ranking Priority Profile (RPP) [17], Xie et al. [18] proposed to use the two-dimensional Singular Spectrum Analysis with the Weighted Median Filter in the design of the cost function, which provides comparable performance to the state-of-the-art yet being relatively computational-efficient.

After determining the adaptive steganography, post-processing techniques are applied to further improve the security of steganographic methods, i.e., [19, 20, 21, 22, 23]. For example, Li et al. [19] proposed the clustering modification directions (CMD) to exploit the interactions among embedding changes, which could decompose the cover image into multiple sub-images and dynamically adjust the cost of pixels in these images. Recently, Chen et al. [20] proposed to modify the generated stego images based on the residual distance between the cover and the modified stego images rather than the cost maps.

Due to the fast development of the deep Convolutional Neural Network (CNN), many tasks in pattern recognition are greatly improved. In image steganalysis, taking the stego signals as a kind of pattern has facilitated various CNN-based steganalysis models, such as the Steganalysis Residual Network (SRNet) [24], Siamese Steganalysis Network (SiaStegNet) [25], and the Global Covariance Pooling Network [26], which is referred to as Deng-Net in this paper.

Table 1: Comparison of different post-processing methods.

Method	Modifications on	Elements used to update
CMD [19]	Stego Images	Costs, directions of pixels
Chen et al. [20]	Cover Images	Stego Images
Zhou et al.[21]	Cover Images	Costs, pixels
Song et al. [27]	Cover Images	Costs, Signs of Gradients
Liu et al. [28]	Cover Images	Costs, Signs of Gradients
Proposed method	Cover Images	Costs, Signs and Magnitudes of Gradients

In addition, CNN is also widely applied in image steganography, where a GAN [29] is often adopted [30, 31, 32, 33, 34, 35, 36, 37]. For example, the Automatic Steganographic Distortion Learning Framework with GAN (ASDL-GAN) [30] is designed to learn an embedding probability directly from a given cover image, which requires training two networks, i.e., a steganographic generative network and a steganalytic discriminative network. As an enhanced ASDLGAN, the UT-GAN [31] is much faster and more powerful in securing the embedded messages. However, the Adversarial Embedding (ADV-EMB) aims to embed the secret message into the cover image while fooling the CNN steganalyser. In [33], the cost function is built iteratively along with a min-max strategy after each iteration.

Recently, Song et al. [27] combines both the cost maps from a steganographic algorithm and the gradients from a pre-trained CNN to adjust the costs and re-generate different stego images. These re-generated stego images will be further compared and selected according to the Manhattan distance between the cover residual and the regenerated-stego residual using the method from [20]. By using the signs of the gradients in the design of the cost function, the security of the stego images has been greatly improved.

However, Song's method does not consider the gradient sub-maps from multiple sub-nets architectures of the CNN-based steganalysis. These gradient sub-maps might have a boundary problem, i.e., unwanted gradients shown in the boundaries of the maps, which may fail to provide satisfactory performance. In their design, only the signs of the gradients are used, whereas the magnitudes of the gradients are ignored, which might not provide enough indications in selecting the suitable embedding areas. The differences among the post-processing techniques are shown in Table 1.

To tackle the previously mentioned boundary problem, in this paper, we propose a novel gradient-guided post-processing method for adaptive image steganography. The major contributions are highlighted as follows.

- A novel gradient-guided post-cost-optimization method is proposed, which considers both the magnitude and the sign of the gradient maps to indicate the embedding positions. In our experiments, it is observed that gradient maps are also capable of indicating peaks and valleys of the magnitude, a useful clue for indicating the high-cost and low-cost areas.

- Within our proposed approach, the previously mentioned boundary problem has been successfully solved.

- The curriculum training strategy of the current CNN-based steganalysers is also investigated, which is often omitted in the previous works. Compared to training from scratch, curriculum training may lead to a different performance of the detectors. However, in our experiments, the situation is fully investigated including tuning the algorithm.

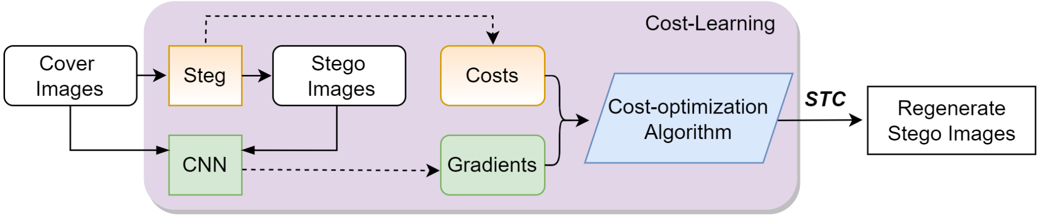

Figure 1: The flowchart of the re-generating stego images, where the Steg is short for Steganography, and the dash lines mean the production process.

The remainder of the paper is organized as follows. Section 2 introduces the background knowledge and the related works. In Section 3, the details of the proposed algorithm are given. Experimental settings and results are presented in Section 4, where an ablation study is also provided. Finally, some concluding remarks are drawn in Section 5.

2. Related Works

In this section, the research background and related works are introduced. We will first explain how adaptive image steganography is modelled in a framework of distortion minimization. Next, the recently proposed Song et al.'s post-processing method with gradients is analysed [27]. Lastly, as the post-cost-optimization method will generate multiple stego images for each cover image, a selection process to choose the best stego images will be discussed.

Let $\mathbf{C}$ and $\mathbf{S}$ denote an 8-bit grey cover image and its stego image, and C_{ij}, S_{ij} represent their pixels in the ith row and jth column, respectively. We have $\mathbf{C} = (C_{ij}), \mathbf{S} = (S_{ij}) \in \{0, \ldots, 255\}^{n_1 \times n_2}$, where n_1 and n_2 denote the width and height of the image, respectively. The superscript k will be used to represent the element in a set $\mathcal{C}$, i.e., the kth cover image in the cover image set, $\mathbf{C}^k \in \mathcal{C}$.

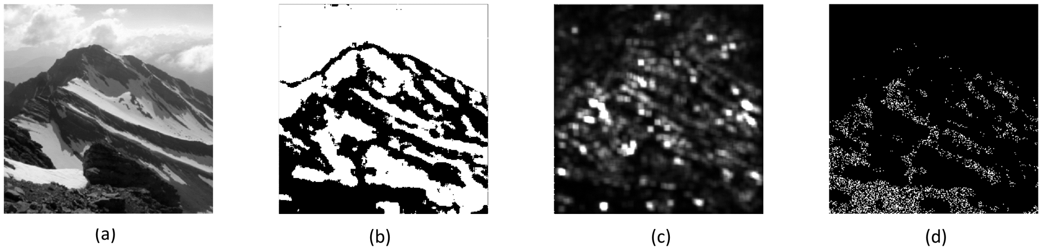

Figure 2: An example to show (a) the cover image, (b) its processed cost map, (c) its gradient map, and (d) its embedding areas created by the proposed method.

2.1. Adaptive image steganography

In Filler et al. [38], the artefacts caused by embedding to each pixel are assumed to be independent of each other, thus every element in the cover image can be assigned with a scalar to indicate the cost of modifying it.

As the image steganography aims to provide secure communication of the embedded message, the total distortion or the cost caused by the embedding needs to be as small as possible. The impact of the embedding, i.e., the modifications to the cover image, is measured by a distortion function $\mathbf{d}(\mathbf{C}, \mathbf{S})$ below, where $\varrho_{ij} \geq 0$ denotes the cost or the security expenditure of changing the pixel value from C_{ij} to S_{ij}.

$$\mathbf{d}(\mathbf{C}, \mathbf{S}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \varrho_{ij}(C_{ij}, S_{ij}) \left| C_{ij} - S_{ij} \right| \tag{1}$$

With the determined embedding cost, the pixel C_{ij} can be designated for embedding with a probability P_{ij}:

$$P_{ij} = \frac{e^{-\lambda \varrho_{ij}}}{1 + 2e^{-\lambda \varrho_{ij}}} \tag{2}$$

where the Lagrange multiplier $\lambda > 0$ is determined from the payload constraint (for the payload-limited sender) by:

$$\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} H(P_{ij}) = m \tag{3}$$

In (3), m is the total number of bits to be embedded and $H(x) = -2x \log x - (1 - 2x) \log(1 - 2x)$ denotes the ternary entropy embedded at P_{ij} [39]. Given a cover image, after calculating its probability map $\mathbf{P}$, its stego image can be created using some near-optimal coding schemes, e.g. Syndrome-Trellis Codes (STCs) [40] [41], to complete the embedding in adaptive steganography.

2.2. Concept of post-cost-optimization

Aiming to enhance the current adaptive steganography via stego generation and selection, Song et al. [27] proposed a post-processing method, where the whole framework for re-generating the stego images is given in Fig. 1. First, a set of cover images and a steganographic algorithm are selected, where the corresponding stego images are created using the steganographic algorithm. At the same time, the cost maps for each of these cover images are determined. Next, the cover and stego image pairs are used to train a CNN, where the gradient map for each of these cover images is produced from the trained CNN. The gradient map and the cost map from the steganography are then utilized for optimizing the cost. The new cost maps will be used to re-generate the stego images.

In Fig. 1, the functional module of Cost-Learning is usually replaced by a GAN. Currently, most of the previous arts consider using the GAN to generate the cost map. For example, papers [30, 31, 32, 34, 37]. However, some papers are trying to reach the end of generating a new stego image directly [35, 36]. To the best of our knowledge, the previous arts using GAN to generate the cost map provide better performance. Hence, we followed the direction and provide our solution.

In Song et al.'s approach [27], for a given cover image, the gradient matrix $\mathbf{G}$ in the same size as the cover is generated from a pre-trained CNN. Let the superscripts $+$ and $-$ denote the modification of the pixel value by plus one and minus one of the pixels, respectively, the cost matrices from a specific steganographic algorithm, i.e., HILL for example, can be written as ρ^+ and ρ^-. Let ϱ^+_{ij} denote the embedding cost at position (i, j), and $\alpha > 1$ denote the adversarial intensity. For the cost map, if the gradient value of the pixel is negative, the corresponding cost value of the pixel remains the same, otherwise, it is increased by the adversarial intensity α. A candidate stego image can be created using the new cost matrices ϱ^+ and ϱ^- as follows, where G_{ij} represents the gradient value of the pixel in the ith row and jth column.

$$\varrho^+_{ij} = \begin{cases} \rho^+_{ij}, & G_{ij} < 0 \\ \rho^+_{ij} + \alpha, & G_{ij} > 0 \end{cases} \tag{4}$$

$$\varrho_{ij}^{-} = \begin{cases} \rho_{ij}^{-} + \alpha, & G_{ij} < 0 \\ \rho_{ij}^{-}, & G_{ij} > 0 \end{cases} \tag{5}$$

2.3. Stego image selection

With the post-cost-optimization algorithm, for each cover image, a set of N_S stego images will be generated for further selection. In image steganalysis, the image residuals after high-pass filtering are the key to differentiating the cover and the stego images. Hence, the distances of the residuals between a cover image and its stego images should also be considered when selecting the best re-generated stego samples. To this end, the residual distance in [20] is used in post-processing of the stego images. Moreover, this process is further adopted in the selection process in [27] as briefed below.

Let $\mathbf{C}^k$ denote a cover image in the cover image set $\mathcal{C}$ with N_C samples, $\mathbf{C}^k \in \mathcal{C}, k = 1, \ldots, N_C$. For $\mathbf{C}^k$, let $\mathbf{S}^{k,0}$ denote the original stego image created by the steganography, a residual function $\mathcal{F}_R(x)$ is employed to the cover image and all its stego images are denoted as $\mathbf{S}^{k,0}$, $\mathbf{S}^{k,1}$, ..., $\mathbf{S}^{k,l}$, ..., $\mathbf{S}^{k,N_S}$, yielding a serious of residuals of $\mathcal{F}_R(\mathbf{C}^k)$ and $\mathcal{F}_R(\mathbf{S}^{k,0})$, $\mathcal{F}_R(\mathbf{S}^{k,1})$, ..., $\mathcal{F}_R(\mathbf{S}^{k,l})$, ..., $\mathcal{F}_R(\mathbf{S}^{k,N_S})$. These residuals are created by three adaptive high-pass filters B_i inspired by [42]. The size of the filters is 7, which is experimentally validated in [20]. Last, the Manhattan distances $\mathcal{F}_D$ between $\mathcal{F}_R(\mathbf{C}^i)$ and all the residuals of the stego images are calculated, where the stego image with the smallest distance will be selected [27].

$$\min \mathcal{F}_D(\mathcal{F}_R(\mathbf{C}^k), \mathcal{F}_R(\mathbf{S}^{k,l})) \tag{6}$$

$$\mathcal{F}_R(x) = \sum_{i=1}^{3} x \otimes B_i \tag{7}$$

3. The Proposed Method

The proposed method is inspired by several observations. First, often, a higher value is observed from the cost map when the magnitude in the gradient map is small. The reasons are mainly two fold. A large magnitude is a result of the “wet costs” [43] or high-risk areas, which prevents the STC from embedding into these areas. Hence, these areas are usually assigned with an extremely large cost, or the wet cost, i.e., 10e+8, while the magnitudes of the costs in the suitable areas are usually less than 1. The second reason is that an effective CNN is usually equipped with a Softmax layer, which maps the magnitude of the output to a normalized interval of $[-1, 1]$ before data classification. The large jump of the magnitude needs to be considered carefully during the design of the new cost map.

Next, the magnitude of the gradient map is also important, in addition to the sign of the gradient as used in Song et al.'s approach [27]. If a pixel is assigned with a large gradient, this pixel seems more important for the prediction. Hence, pixels with large magnitudes in a gradient map should be carefully processed for improving the performance of steganography.

3.1. Process the gradient

As different CNNs have various network architectures, the input cover images are processed in different ways. For example, some CNNs contain multiple sub-nets for parameter-optimization or improved efficiency in training [25]. To smooth the boundaries of the sub-maps of the gradients created by such CNNs, often a low-pass filter is used. According to the RPP [17] [18], during the embedding, the embedding areas should better be clustered to avoid being easily detected, hence an improved security performance. This clustering process can be realized by a low-pass filter as detailed below.

Let $\mathbf{G}^k$ be the gradient matrix generated from a pre-trained CNN, F, for the cover image $\mathbf{C}^k$, we have $\mathbf{G}^k = F(\mathbf{C}^k)$. Denote $\mathbf{L}_r$ as the average filter with a kernel size r. For a cover image $\mathbf{C}^k$, we can obtain a gradient matrix $\mathbf{g}^k$ below, where $\mathbf{g}^k$ and $\mathbf{G}$ are of the same size.

$$\mathbf{g}^k = \left|\mathbf{G}^k \otimes \mathbf{L}_r\right| \tag{8}$$

3.2. Cost map selection

Let $\rho^{k,+}$ denote the cost matrix of increasing the pixel value of $\mathbf{C}^k$ by one and $\rho^{k,-}$ the cost matrix of decreasing its pixel value by one. Both $\rho^{k,+}$ and $\rho^{k,-}$ are from the steganography Φ. We can rewrite the $\rho^{k,+}$ and $\rho^{k,-}$ as in (9), where $N = n_1 \times n_2$, and $\rho_1^{k,+/-} \leq \rho_2^{k,+/-} \leq \cdots \leq \rho_N^{k,+/-}$.

$$\rho^{k,+/-} = \sum_{j=1}^{N} \rho_j^{k,+/-} \tag{9}$$

Define a selecting interval θ, $\theta = [\theta_l, \theta_h]$, where θ_l indicates the lower bound and θ_h the upper bound. We can choose the pixels of the desired costs within the selecting interval, by:

$$\varrho_\theta^{k,+/-} = \sum_{j=1}^{N} \delta(\rho_j^{k,+/-}), \quad 0 \leqslant \varrho_\theta^{k,+/-} \leqslant N \tag{10}$$

$$\delta(\rho_j) = \begin{cases} 1, & N \cdot \theta_l \leq j \leq N \cdot \theta_h \\ 0, & else \end{cases} \tag{11}$$

Here, we map the large costs, i.e., ranking $N \cdot \theta_h$ to N, to 0 and map the smaller costs to 1 for further processing below.

3.3. Generate the new cost map

Let β_g denote the adversarial intensity, we can calculate the new cost map $\varrho^{k,+}$ based on the gradient map $\mathbf{g}^k$ and the modified cost map $\varrho_\theta^{k,+}$ as follows:

$$\varrho^{k,+} = \left|1 - \varrho_\theta^{k,+} - \beta_g \cdot \mathbf{g}^k\right| \tag{12}$$

$$\varrho^{k,-} = \left|1 - \varrho_\theta^{k,-} - \beta_g \cdot \mathbf{g}^k\right| \tag{13}$$

The formulas can be explained in this way. First, ensure the magnitude in the cost maps are no longer the dominant factors, they are mapped to $\{0, 1\}$ using (11). To adjust the extreme large magnitude of the wet costs from the previous cost map ρ, these costs will be mapped to 1 by $1 - \varrho$, where ϱ has already mapped the wet costs to 0. Notice that in $1 - \varrho$, the small costs will be mapped to 0. Now the small-cost areas have the same weights. To accurately guide the embedding process, the magnitudes of the elements in the gradient map are employed. Although the magnitudes in the gradient map are small, they are capable of indicating the peaks and valleys, or the relatively high-cost and low-cost areas.

As illustrated in Fig. 2, the cover image '472.pgm' in the BOWS dataset is shown in (a), along with its processed cost map $1 - \varrho_\theta^+$ shown in (b), its gradient map $\mathbf{g}$ shown in (c), and the embedding areas in (d). In the processed cost map, the white pixels represent 1 and the black ones represent 0, where those white pixels are not allowed to embed due to the large associated costs. In the gradient map, the overall magnitude is small. However, it does provide the focused areas for embedding by adding weights to the cost map. Hence, the exact locations are determined by the gradient map. As $\varrho^{+/-}$ is non-negative, an absolute operator is applied here.

3.4. Dealing with the wet costs

To ensure that easy-to-spot pixels in the cover image are not used for embedding, a wet cost, i.e., 10e+8, needs to be defined. Let $\varrho_{ij}^{k,+}$ and $\varrho_{ij}^{k,-}$ be the cost values in the ith row and jth column in $\varrho^{k,+}$ and $\varrho^{k,-}$, respectively, we can adjust the corresponding cost value as follows:

$$\begin{aligned} \varrho_{ij}^{k,+} &= 10e+8, \quad if \quad \mathbf{C}_{ij}^{k} = 255 \\ \varrho_{ij}^{k,-} &= 10e+8, \quad if \quad \mathbf{C}_{ij}^{k} = 0 \end{aligned} \tag{14}$$

In this way, these pixels are ensured to avoid being candidates for embedding.

3.5. Generating multiple stego samples

By adjusting the selecting interval θ, a set of N_S stego images can be generated. The most suitable one will be selected based on Eqs. (6) and (7). Finally, the whole framework of generating stego images is summarized in Algorithm 1.

4. Experimental results and analysis

4.1. Experimental settings

4.1.1. Datasets

The widely used BOSSbase v1.01 [44] and BOWS2 [45] datasets are used in our experiments, and each contains 10,000 uncompressed images sized of 512×512 pixels. All the images are resized to 256×256 using the *imresize()* function in MATLAB. The stego images are created from the following adaptive steganographic methods, S-UNI [8], HILL [9] and WOW [1]. The relative payloads tested are 0.1, 0.2, 0.3 and 0.4 bpp, respectively.

For a specific payload, the whole dataset is evenly divided into two non-overlapping parts at random. The first half is used to train the CNN and create the gradients for the whole dataset. The second half is used to re-train the CNN and test the security performance, with 5000 cover-stego pairs used to re-train the model and the remaining 5000 pairs for evaluation.

Algorithm 1 The proposed stego image regeneration algorithm

```
Input: A set of $N_C$ cover images $\mathbf{C}^k \in \mathcal{C}$, the original stego image $\mathbf{S}^{k,0}$, the gradient map $\mathbf{g}^k$ and the cost maps: $\rho^{k,+}$ and $\rho^{k,-}$
Output: A set of $N_C$ stego images $\mathbf{S}^k$
for $k = 1$ to $N_C$ do
    for $l = 1$ to $N_S$ do
        $T = \mathcal{F}_D(\mathcal{F}_R(\mathbf{C}^k), \mathcal{F}_R(\mathbf{S}^{k,0}))$
        Generate $\mathbf{S}^{k,l}$ according to (8) to (14) at the same payload as $\mathbf{S}^{k,0}$
        if $\mathcal{F}_D(\mathcal{F}_R(\mathbf{C}^k), \mathcal{F}_R(\mathbf{S}^{k,l})) < T$ then
            $T = \mathcal{F}_D(\mathcal{F}_R(\mathbf{C}^k), \mathcal{F}_R(\mathbf{S}^{k,l})$
        end if
    end for
    if $\mathcal{F}_D(\mathcal{F}_R(\mathbf{C}^k), \mathcal{F}_R(\mathbf{S}^{k,l})) < T$ then
        Return $S^k = S^{k,l}$
    else
        Return $S^k = S^{k,0}$
    end if
end for
```

4.1.2. The settings of the CNNs

Two classic CNNs for image steganalysis, i.e., the SiaStegNet [25] and the Deng-Net [26] are utilized to generate the gradients. This is because both of them can provide SOTA performance, and Deng-Net represents the CNN with only one network while the SiaStegNet has two sub-nets. The hyperparameters are all kept the same as defaults as detailed below. For the SiaStegNet, the Adamax optimizer [46] with an initial learning rate set to 0.001, and $\beta_g = [0.9, 0.999]$ is used. For the Deng-Net, the optimizer Stochastic Gradient Descent (SGD), is used with a momentum of 0.9. We set both CNNs to the default initialization method during the training.

Curriculum training [24] is used when re-train for payloads below 0.4 bpp, as widely adopted by most CNNs for improved performance [24, 25]. For the SiaStegNet, except for the 0.4 bpp scenarios where the training is run for 500 epochs, all the curriculum training will run for 200 epochs for fine-tuning.

While for the Deng-Net, except for the 0.4 bpp cases where epochs are set to 200, the network is trained for 100 epochs for fine-tuning.

Data augmentation was employed for all CNNs, which include random rotation for 90 degrees and random flip with a probability of 0.5. The batch size for all the CNNs is set to 32 by default. All the experiments were carried out with Pytorch 1.7.1 on a Tesla V100 Graphics Processing Unit.

4.1.3. Settings of other parameters

In Song et al.'s post-processing algorithm, all the settings are set to default, where the adversarial intensity remains to be $\alpha = 2$, and the number of the generating stego sample is $N_S = 100$.

For a fair comparison, we set the $N_S = 100$ in our method as default. We set the adversarial intensity as $\beta_g = 0.025$ and the kernel size $r = 7$. The selecting interval θ is created using a continuous uniform random number generator, with the lower endpoint θ_l set to 0.1, and the upper endpoint θ_h to 0.5 by default. The optimized parameters for N_S, θ_l, θ_h are $40, 0.2, 0.8$, which will be justified below.

4.2. Ablation Study

4.2.1. Modification areas among different methods

We first show a cover image with edges and details, the '472.pgm', its embedding areas using the SUNI algorithm, Song et al.'s method and our method in Fig. 3 for comparison, to compare the differences of the embedding. Fig. 3 (c-d) are the embedding areas in the regenerated stego images that have been successfully selected by Algorithm 1.

As shown in Fig. 3 (c), although Song's method had successfully created a new stego image, the embedding area is very similar to the SUNI's. However, our method generated significantly different embedding areas, which are more clustered than the other two, indicating the efficacy of the proposed low-pass filter in smoothing the noise.

4.2.2. Differences of the gradients

One may ask why the gradients from the conventional methods are not used but the ones from the CNNs. To answer this question, we show the image gradients computed by using different methods in Fig. 4. These gradients are generated by simply replacing the F with the gradient operators, such as 'Sobel' and 'Roberts'. The 'Sobel' and 'Roberts' are the operators that are used to emphasise the edges in the images. We show the horizontal gradients

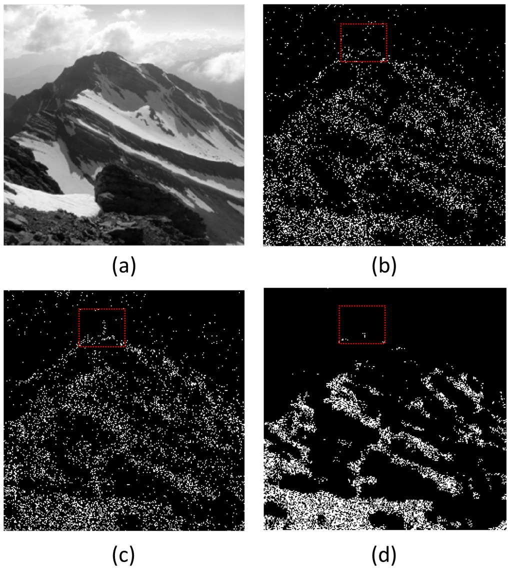

Figure 3: The comparisons of embedding areas in different methods at 0.4 bpp. (a) cover image, (b) original SUNI method, (c) Song et al.'s method, and (d) the proposed method.

and the gradients along 45 degrees. The vertical gradient and the 135-degree gradients are not shown due to the limited space, yet the conclusions remain the same.

In Fig. 4, there's almost no difference among them, which indicates that simply using the gradients directly from these traditional methods can not capture the weak stego signals. We also calculated the difference between the cover image and stego images, again nothing noticeable was found.

Alternatively, we show the gradient maps from two CNNs in Fig. 5 and Fig. 6. As seen, the gradient maps generated from the CNNs are different from each other, though the images are visually the same. The magnitudes of the gradient maps are different as well. For example, the maximum value of the cover image from the Deng-Net is 0.0477 and the minimum is 9.09e-10; whilst the maximum value of the stego image from the Deng-Net is 0.1396 and the minimum is 1.07e-9. In addition, the maximum value of the cover image from the SiaStegNet is 0.0033 and the minimum is 2.8e-11; the maximum value of the stego image from the SiaStegNet is 0.4713 and the minimum is 2.37e-9. The large difference in the maximum value between the cover and stego image helps the CNNs to differentiate the two images.

4.2.3. The influence of the adversarial factor β_g and the kernel size r

Here, we will analyze the influences of the adversarial factor β_g and the kernel size r to see how they affect the security of the stego images. The generated stego images will be retrained using the same Deng-Net with the same settings, and the results are shown in Table 2.

As seen in Table 2, the best result is achieved with $\beta_g = 0.025$ and $r = 7$, where further increasing or decreasing β_g may result in a degraded result. Moreover, decreasing the kernel size from 7 will cause about 1% performance loss. This may suggest that under the current settings, this kernel size works the best.

Table 2: The influence of the adversarial intensity β_g and the kernel size r.

β_g	r	Retrain Acc (in %)
0.0125	5	77.04
0.025	5	77.23
0.05	5	76.94
0.0125	7	77.12
0.025	7	**76.25**
0.05	7	76.70
0.0125	9	76.49
0.025	9	76.31
0.05	9	76.61

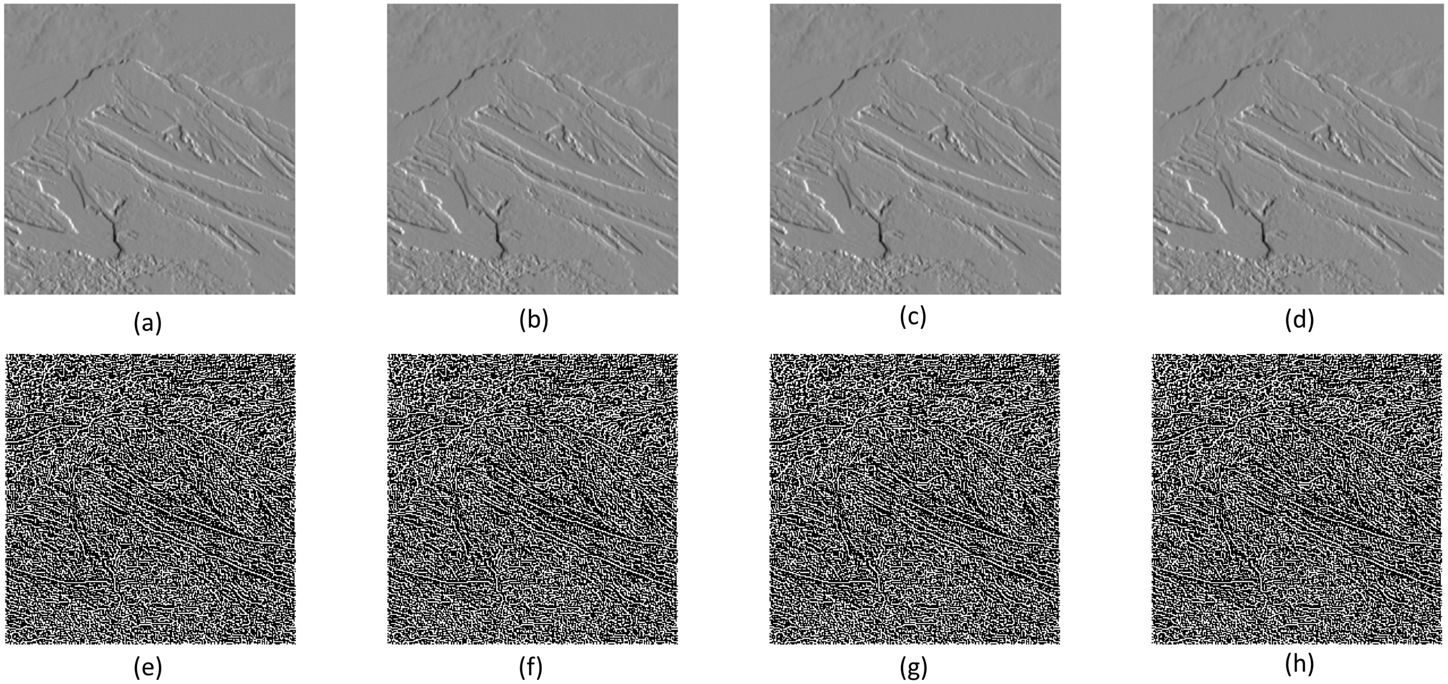

Figure 4: Simply using image gradients from the conventional method will not differentiate the cover and stego images. Fig. (a) to (d) are created by the ‘Sobel’ operator; (a) cover image, (b) the original SUNI stego image, (c) Song et al.’s stego image, (d) stego image of the proposed method. Fig. (e) to (h) are created by the ’Roberts’ operator; (e) cover image, (f) original SUNI stego image, (g) Song et al.’s stego image, (h) stego image of the proposed method.

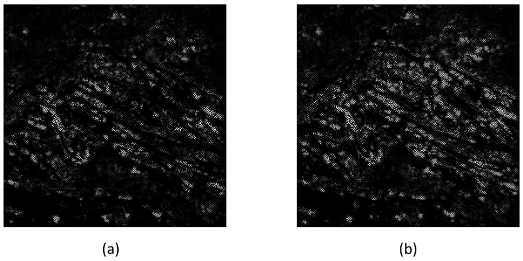

Figure 5: Gradient maps from the Deng-Net with gradients enhanced by 500 times. (a) the cover image; (b) stego image.

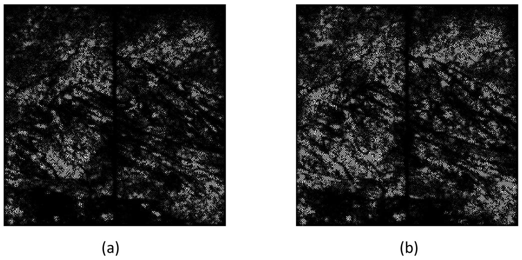

Figure 6: Gradient maps from the SiaStegNet. (a) cover image (enhanced by 500 times); (b) stego image (enhanced by 50 times).

Table 3: Retrain accuracy of different numbers of generating stego samples N_S.

N_S	10	20	30	40	50	60	70	80	90	100
Retrain Acc (in %)	77.14	76.89	76.67	**76.30**	76.74	76.51	76.47	76.60	76.58	76.36

Table 4: Retrain accuracy of different selecting intervals θ.

$\theta = [\theta_l, \theta_h]$	[0.1, 0.4]	[0.1, 0.5]	[0.2, 0.5]	[0.1, 0.6]	[0.1, 0.7]	[0.1, 0.8]	[0.1, 0.9]	[0.2, 0.8]	[0.3, 0.8]
Retrain Acc (in %)	77.63	76.30	76.88	75.78	74.85	74.72	75.03	**74.56**	75.18

4.2.4. The number of generated stego samples

To reduce the time in generating and selecting the stego samples, the best candidate for the number of generated stego samples needs to be determined. For this purpose, only the number N_S varies and the detection accuracies of retraining those images are shown in Table 3. The results are obtained by retraining the SiaStegNet on the regenerated HILL stego samples at 0.4 bpp.

In Table 3, it is observed that 40% of the generated samples can provide a similar result to the default setting of 100 samples. Also, reducing the number from 40 will deteriorate the security performance.

4.2.5. The influence of different selecting-intervals θ

To determine the best selecting-interval θ in the proposed algorithm, one of the parameters θ_l and θ_h will be changed each time and the resulting stego samples will be re-generated. Then, the SiaStegNet will be re-trained at 0.4 bpp, just as in the last experiment. The detection accuracies of retraining those images are reported in Table 4.

Starting from [0.1, 0.5], θ_h is decreased by 0.1 and the performance drops about 1%. Increasing θ_l yields a similar result. Hence, θ_h is progressively increased by 0.1 and the result is getting better until θ_h reaches 0.9. In this way, the best candidate for θ_h is found to be 0.8, and the θ_l is evaluated by progressively increasing it by 0.1. Finally, the best result is found when $\theta_l = 0.2$ and $\theta_h = 0.8$, which is about 2% better than the default setting.

Table 5: Detection Accuracy (%) on Three Re-Trained Steganalyzers. Ori means Original steganography algorithm; Prop (Def) means the Proposed method with Default settings and Prop (Opti) means the Proposed method with Optimized settings.

		SiaStegNet				Deng-Net				SRM			
Steganography	Payload (bpp)	Ori	Song et al.	Prop (Def)	Prop (Opti)	Ori	Song et al.	Prop (Def)	Prop (Opti)	Ori	Song et al.	Prop (Def)	Prop (Opti)
WOW	0.4	87.88	86.22	77.73	**74.93**	88.58	85.50	81.32	**80.01**	74.96	74.77	71.50	**69.24**
	0.3	83.57	81.35	70.51	**68.17**	85.82	81.29	78.07	**74.72**	70.40	69.56	64.89	**63.64**
	0.2	78.30	75.42	62.55	**61.89**	79.18	73.89	70.29	**68.53**	63.59	62.98	**58.28**	58.38
	0.1	68.60	65.55	**56.57**	59.36	69.30	63.13	**59.24**	60.91	56.22	55.74	**53.30**	54.37
SUNI	0.4	86.80	84.36	77.47	**74.95**	86.05	83.37	**79.54**	79.96	74.92	74.34	71.56	**70.10**
	0.3	81.67	78.55	70.02	**68.24**	82.15	78.10	74.13	**72.68**	69.20	68.47	64.91	**64.21**
	0.2	75.30	72.03	62.60	**62.25**	74.85	70.11	66.87	**66.57**	63.03	62.40	59.02	**58.98**
	0.1	64.10	61.49	**55.10**	56.22	63.93	61.06	**57.97**	59.30	55.67	55.09	**53.23**	54.23
HILL	0.4	82.00	79.55	76.36	**74.56**	81.80	76.82	76.25	**75.29**	69.54	69.32	68.44	**67.34**
	0.3	77.60	74.13	69.76	**68.48**	76.35	72.51	72.08	**70.47**	64.53	64.35	62.47	**61.98**
	0.2	71.78	68.32	**62.30**	62.71	71.25	67.57	65.96	**64.98**	59.37	58.80	**56.48**	56.91
	0.1	63.10	59.89	**55.22**	55.90	62.35	61.58	58.59	**58.01**	53.60	53.27	**52.46**	52.76

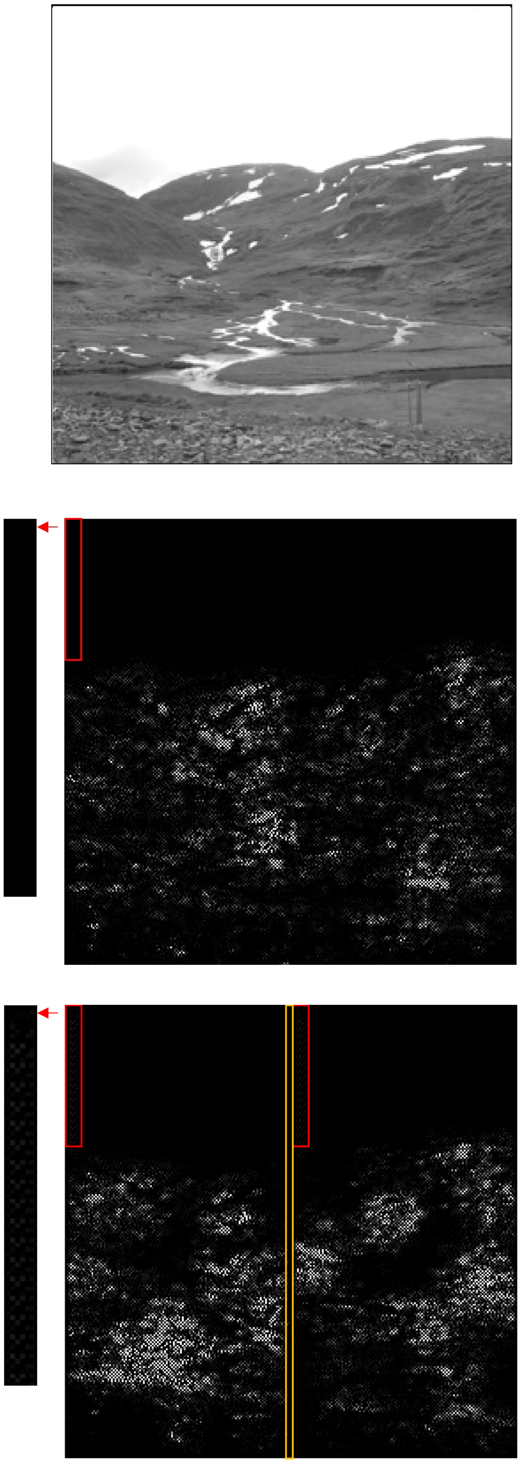

Figure 7: The comparisons of image gradients produced from different CNNs: Cover image (top), Deng-net (middle), and SiaStegNet (bottom). The red rectangular areas in the gradient maps are enlarged on the left of them.

4.3. Performance evaluation

We compare our method with other steganographic methods against different steganalysis techniques and show the results in Table 5. For the conventional steganalysis methods, the Spatial Rich Model (SRM) [47] is employed to provide the detection results, which considers the quantized image noise residual and its distribution. The cover and stego images are used to train Ensemble classifiers [48], which are capable of detecting the stego noises in the stego images. To avoid confusion, the experimental results of the SRM in this table are created with samples generated using the gradients from the SiaStegNet.

Starting from the steganalysis results of the SiaStegNet, it is obvious that our proposed methods provide the best performance among all three steganographic algorithms. For WOW, Song et al.'s method provides an improvement of up to 3% across four payloads while ours can reach a 12% improvement. For SUNI, the situation is about the same as WOW. However, the improvement achieved by our method is slightly smaller due to the higher security of the original SUNI algorithm. The improvement achieved by our method is even smaller for the steganographic algorithm HILL, though still the best among all the compared algorithms.

Another observation is that when using optimized settings, our method can provide further improvements when the payload is 0.2 bpp or larger. For the scenarios of 0.1 bpp payload, it is suggested to use the default settings. This is because, in an extreme low payload situation, the number of the embedding areas that allow the algorithm for selecting is small, hence requiring N_S to be large enough to create more samples for further selection.

For the results from the Deng-Net, some observations are highlighted below. First, although Deng-Net provides a similar steganalysis performance to the SiaStegNet, the security performance provided by the Song et al.'s method is improved in most cases. However, the security performance provided by our method is not as good as using the gradients from the SiaStegNet, especially for the WOW algorithm. This might be due to the multi-subnet architecture of the SiaStegNet, whose gradient sub-maps are diverse enough for our algorithm to create different re-generated samples.

Nevertheless, our proposed methods are still superior to Song et al.'s method in all steganographic methods under different payloads. The margins remain large especially when the embedding payloads are 0.1 and 0.2 bpps. Again, the proposed method with the optimized settings is a better option when the payloads are 0.2 and 0.3 bpps. For the WOW and SUNI, the default settings are still the best. For HILL, the optimized settings achieved the best performance for every payload listed.

For the conventional methods, i.e., the SRM attack [47], Song's method has a limited improvement while ours with the default settings can still achieve about 3% improvement in the low payload scenarios on average except for the HILL. For payloads 0.3 and 0.4 bpps, the optimized settings are the best selection.

To further explain these results, we draw the gradients generated using the two different CNNs in Fig. 7 with some observations highlighted below. First, the gradient maps shown can indeed indicate the edges in the cover image. Second, for this cover image, the gradient map from the SiaStegNet is more clustered than the one from the Deng-Net.

The gradient map, shown in Fig. 7 (c), is a result of two sub-images due to the sub-net architecture of the SiaSteNet. The orange rectangular area separates the left and the right gradient map. This has led to two issues. First, the red rectangular areas in the top-left show the faked gradients, and this may have misled Song et al.'s method to select these areas for embedding. However, thanks to our proposed low-pass filter, these false-alarm areas have been successfully removed. Next, due to the hard separation part in the middle of the gradient image, this has inevitably introduced problems in the weight-ranking process of Song et al.'s method.

One last observation from the results of the SiaStegNet is the proposed method with the optimized settings has achieved comparable results for three different steganographic algorithms. This can be explained in Fig. 8, where the embedding areas in Figs. 8 (b)-(d) are scattered compared to the ones created by the proposed algorithm with optimized settings. In Figs. 8 (f)-(h), all three images indicate the clustering effect of the proposed algorithm, much different from the original HILL algorithm, which may explain why they are less vulnerable to the attacks [19].

4.4. Security performance of stego samples created by different gradients against SRM attack

In Table 5, we have shown the security performance of the stego samples produced using the gradients from the SiaStegNet against SRM. For comparison, we also show the corresponding results using the gradients from the Deng-Net in Table 6.

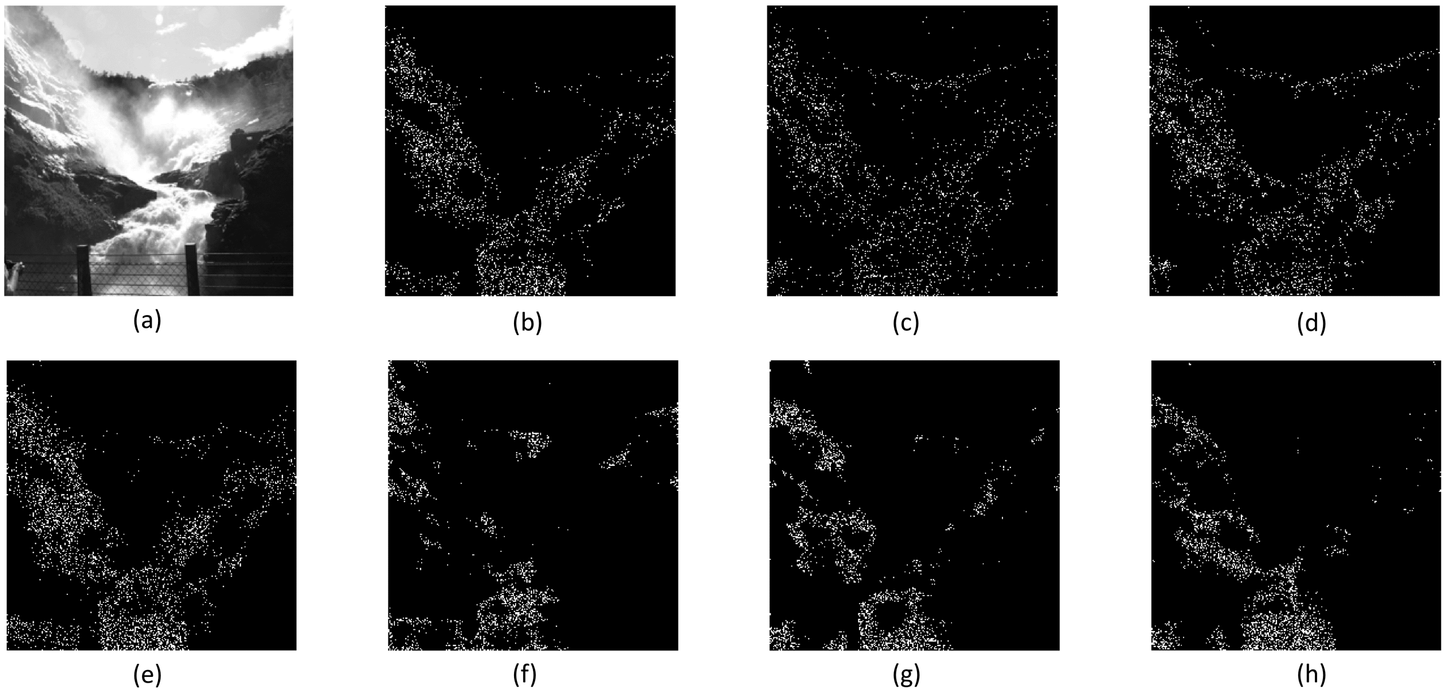

Figure 8: Comparisons of embedding areas among different algorithms at 0.3 bpp. (a) Cover image; (b-d) WOW, SUNI, HILL with Song et al.'s post-processing algorihtm; (e) Original HILL algorihtm; (f-h) WOW, SUNI, HILL with the proposed post-cost-optimization algorithm.

As seen in Table 6, both Song et al.'s approach and the proposed method can improve the performance from the original steganographic methods. However, the margins between them are small. The reasons are mainly two fold. First, the gradient map generated from the Deng-Net is a complete map instead of two split parts, which will make the gradient easier to process for Song et al.'s method. Second, the gradients created from the Deng-Net are more compact than the corresponding components from the SiaStegNet's, hence the lowpass filter in our method can only help more for the WOW and SUNI.

Table 6: Detection Accuracy (%) of SRM for the stego images created using Deng-Net's gradients (The results are averaged for 3 times).

		SRM*			
Steg	Payload (bpp)	Ori	Song et al.	Prop (Def)	Prop (Opti)
WOW	0.4	74.96	73.58	73.12	**72.56**
	0.3	70.40	68.47	68.08	**67.22**
	0.2	63.59	61.73	**61.04**	61.54
	0.1	56.22	54.61	**53.95**	54.90
SUNI	0.4	74.92	73.16	72.58	**72.25**
	0.3	69.20	67.48	66.42	**66.00**
	0.2	63.03	61.23	**60.32**	60.77
	0.1	55.67	54.64	**53.63**	54.69
HILL	0.4	69.54	67.51	68.12	**67.47**
	0.3	64.53	**62.47**	62.96	62.73
	0.2	59.37	58.01	**57.53**	57.96
	0.1	53.76	**52.52**	53.00	53.52

4.5. Modification rate comparisons

In this subsection, we calculate the actual changes to the original stego image with different optimization methods. For this purpose, we define the modification rate R_S in (15) and the average modification rate $\overline{R_S}$ in (16). Given a cover image $\mathbf{C}^k$, we use a given steganographic algorithm, i.e., HILL, to produce the original stego image $\mathbf{S}^{k,0}$. Afterwards, we optimize the cost

Table 7: Average modification rate of different methods.

		SiaStegNet			Deng-Net		
Steg	Payload (bpp)	Song et al.	Prop (Def)	Prop (Opti)	Song et al.	Prop (Def)	Prop (Opti)
WOW	0.4	4.72	2.39	3.74	4.24	1.80	2.42
	0.3	3.33	2.68	3.35	3.25	1.52	2.15
	0.2	2.31	2.03	1.88	2.22	1.42	1.53
	0.1	1.23	0.74	0.37	1.17	0.95	0.80
	Avg	2.90	1.96	2.33	2.72	1.42	1.73
SUNI	0.4	6.60	2.22	3.08	6.61	1.64	1.74
	0.3	4.68	2.38	3.09	4.69	1.40	1.77
	0.2	2.92	1.84	1.97	2.93	1.15	1.13
	0.1	1.33	0.83	0.58	1.32	0.67	0.49
	Avg	3.88	1.82	2.18	3.89	1.21	1.28
HILL	0.4	4.61	2.20	3.51	4.47	1.86	2.53
	0.3	3.52	2.41	3.19	3.44	1.54	1.82
	0.2	2.48	1.94	1.83	2.42	1.17	1.00
	0.1	1.32	0.64	0.40	1.28	0.63	0.51
	Avg	2.98	1.79	2.23	2.90	1.30	1.47

Table 8: Comparison of running time in seconds.

Steg	Song et al.'s [27]	Prop (Def)	Prop (Opti)
WOW	3.05	0.98	**0.35**
SUNI	3.41	1.22	**0.39**
HILL	2.95	1.17	**0.43**

using the gradients and re-generate a new stego image $\mathbf{S}^{k,l}$ with a post-cost-optimization algorithm, i.e., Song et al.'s. We compare the average modification rate $\overline{R_S}$ on the Re-generated datasets, each with 10,000 images, between different methods in Table 7.

$$R_s = (\sum_{i,j=1}^{n_1,n_2} |S_{ij}^{k,0} - S_{ij}^{k,l}|) \times 100 \times (n_1 \times n_2)^{-1} \tag{15}$$

$$\overline{R_s} = \sum_{l=1}^{N_S} R_s \times N_S^{-1} \tag{16}$$

As shown in Table 7, the proposed method with default settings introduces much fewer modifications to the stego image compared to the Song et al.'s method under both situations. In the situation where the gradients of the SiaStegNet are used, 35% fewer modifications are introduced on average for WOW. For SUNI, the average figure is about 53%, less than a half of Song et al.'s. For the Deng-Net, the situation is similar to the SiaStegNet.

However, it is worth noting that fewer modifications to the original stego image do not mean better performance against an attack. This can be observed by combining both Table 7 and Table 5. Take WOW at 0.4 bpp for example, where the proposed algorithm with optimized settings achieves the best performance yet it has more modifications than that with the default settings.

4.6. Comparison of the computational cost

We compare the running time in Table 8 and see how the selection process can speed up our method. All the running times are recorded on an AMD 4800H laptop with 8 cores and 16 GB RAM, which is averaged on 4 different payloads. For a fair comparison, the proposed method with default setting produced 100 stego samples for each cover image, and the numbers were recorded on 10,000 cover images. We also show the running time of our optimized algorithm.

As seen in Table 8, it takes about 3 seconds for the whole process to produce one stego image for the WOW algorithm with Song et al.'s method. Our proposed method, however, is about twice as fast as Song et al.'s method [27] in every steganographic method. With the optimized settings, the proposed algorithm can be further sped up by about 65%.

5. Conclusions

In this paper, a new gradient guided post-processing method is proposed to improve the security of image steganography. The idea is inspired by the observations that there exists a large jump in the magnitude between the gradient map and the cost map from the same cover image, where the magnitude in the gradient map matters even though the overall magnitude is often small. By considering the magnitude in the gradient map, we carefully design a new post-cost-optimization method and use it in generating multiple stego images for a given cover image. The best candidate will be selected by a selection algorithm. Comprehensive experiments have validated the effectiveness of the proposed method. In addition, our proposed method is computationally efficient.

In the future, we will focus more on further enhanced post-processing methods, including the GAN-involved methods. At the same time, more network architectures in image steganalysis will be explored in the future for designing such post-cost-optimization methods.

References

[1] V. Holub, J. Fridrich, Designing steganographic distortion using directional filters, in: International workshop on information forensics and security (WIFS), IEEE, 2012, pp. 234–239.

[2] J. Fridrich, Steganography in digital media: principles, algorithms, and applications, Cambridge University Press, 2009.

[3] P. C. Mandal, I. Mukherjee, G. Paul, B. Chatterji, Digital image steganography: A literature survey, Information Sciences (2022).

[4] T. Muralidharan, A. Cohen, A. Cohen, N. Nissim, The infinite race between steganography and steganalysis in images, Signal Processing (2022) 108711.

[5] Y. Zhang, X. Luo, J. Wang, Y. Guo, F. Liu, Image robust adaptive steganography adapted to lossy channels in open social networks, Information Sciences 564 (2021) 306–326.

[6] Z. Zhu, S. Li, Z. Qian, X. Zhang, Destroying robust steganography in online social networks, Information Sciences 581 (2021) 605–619.

[7] D. K. Sarmah, A. J. Kulkarni, Jpeg based steganography methods using cohort intelligence with cognitive computing and modified multi random start local search optimization algorithms, Information Sciences 430-431 (2018) 378–396.

[8] V. Holub, J. Fridrich, T. Denemark, Universal distortion function for steganography in an arbitrary domain, EURASIP Journal on Information Security 2014 (1) (2014) 1.

[9] B. Li, M. Wang, J. Huang, X. Li, A new cost function for spatial image steganography, in: International Conference on Image Processing (ICIP), IEEE, 2014, pp. 4206–4210.

[10] V. Sedighi, R. Cogranne, J. Fridrich, Content-adaptive steganography by minimizing statistical detectability, IEEE Transactions on Information Forensics and Security 11 (2) (2015) 221–234.

[11] D. Hu, H. Xu, Z. Ma, S. Zheng, B. Li, A spatial image steganography method based on nonnegative matrix factorization, IEEE Signal Processing Letters 25 (9) (2018) 1364–1368.

[12] W. Su, J. Ni, X. Hu, J. Fridrich, Image steganography with symmetric embedding using gaussian markov random field model, IEEE Transactions on Circuits and Systems for Video Technology (2020).

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

[14] T. Pevny, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Transactions on information Forensics and Security 5 (2) (2010) 215–224.

[15] J. Fridrich, J. Kodovskỳ, Multivariate gaussian model for designing additive distortion for steganography, in: International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 2949–2953.

[16] X. Qin, B. Li, J. Huang, A new spatial steganographic scheme by modeling image residuals with multivariate gaussian model, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2617–2621.

[17] B. Li, S. Tan, M. Wang, J. Huang, Investigation on cost assignment in spatial image steganography, IEEE Transactions on Information Forensics and Security 9 (8) (2014) 1264–1277.

[18] G. Xie, J. Ren, S. Marshall, H. Zhao, H. Li, A new cost function for spatial image steganography based on 2d-ssa and wmf, IEEE Access 9 (2021) 30604–30614.

[19] B. Li, M. Wang, X. Li, S. Tan, J. Huang, A strategy of clustering modification directions in spatial image steganography, IEEE Transactions on Information Forensics and Security 10 (9) (2015) 1905–1917.

[20] B. Chen, W. Luo, P. Zheng, J. Huang, Universal stego post-processing for enhancing image steganography, Journal of Information Security and Applications 55 (2020) 102664.

[21] W. Zhou, W. Zhang, N. Yu, A new rule for cost reassignment in adaptive steganography, IEEE Transactions on Information Forensics and Security 12 (11) (2017) 2654–2667. doi:10.1109/TIFS.2017.2718480.

[22] W. Li, W. Zhang, K. Chen, W. Zhou, N. Yu, Defining joint distortion for jpeg steganography, in: Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, 2018, pp. 5–16.

[23] K. Chen, H. Zhou, W. Zhou, W. Zhang, N. Yu, Defining cost functions for adaptive jpeg steganography at the microscale, IEEE Transactions on Information Forensics and Security 14 (4) (2019) 1052–1066.

[24] M. Boroumand, M. Chen, J. Fridrich, Deep residual network for steganalysis of digital images, IEEE Transactions on Information Forensics and Security 14 (5) (2018) 1181–1193.

[25] W. You, H. Zhang, X. Zhao, A siamese cnn for image steganalysis, IEEE Transactions on Information Forensics and Security 16 (2020) 291–306.

[26] X. Deng, B. Chen, W. Luo, D. Luo, Fast and effective global covariance pooling network for image steganalysis, in: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, 2019, pp. 230–234.

[27] T. Song, M. Liu, W. Luo, P. Zheng, Enhancing image steganography via stego generation and selection, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 2695–2699.

[28] M. Liu, W. Luo, P. Zheng, J. Huang, A new adversarial embedding method for enhancing image steganography, IEEE Transactions on Information Forensics and Security 16 (2021) 4621–4634.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[30] W. Tang, S. Tan, B. Li, J. Huang, Automatic steganographic distortion learning using a generative adversarial network, IEEE Signal Processing Letters 24 (10) (2017) 1547–1551.

[31] J. Yang, D. Ruan, J. Huang, X. Kang, Y.-Q. Shi, An embedding cost learning framework using gan, IEEE Transactions on Information Forensics and Security 15 (2019) 839–851.

[32] W. Tang, B. Li, S. Tan, M. Barni, J. Huang, Cnn-based adversarial embedding for image steganography, IEEE Transactions on Information Forensics and Security 14 (8) (2019) 2074–2087.

[33] S. Bernard, T. Pevnỳ, P. Bas, J. Klein, Exploiting adversarial embeddings for better steganography, in: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, 2019, pp. 216–221.

[34] F. Li, Z. Yu, C. Qin, Gan-based spatial image steganography with cross feedback mechanism, Signal Processing 190 (2022) 108341.

[35] L. Li, M. Fan, D. Liu, Advsgan: Adversarial image steganography with adversarial networks, Multimedia Tools and Applications (2021) 1–17.

[36] L. Li, W. Zhang, C. Qin, K. Chen, W. Zhou, N. Yu, Adversarial batch image steganography against cnn-based pooled steganalysis, Signal Processing 181 (2021) 107920.

[37] H. Mo, T. Song, B. Chen, W. Luo, J. Huang, Enhancing jpeg steganography using iterative adversarial examples, in: 2019 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6.

[38] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, IEEE Transactions on Information Forensics and Security 6 (3) (2011) 920–935.

[39] J. Butora, Y. Yousfi, J. Fridrich, Turning cost-based steganography into model-based, in: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, 2020, pp. 151–159.

[40] J. Fridrich, T. Filler, Practical methods for minimizing embedding impact in steganography, in: Security, Steganography, and Watermarking of Multimedia Contents IX, Vol. 6505, International Society for Optics and Photonics, 2007, p. 650502.

[41] W. Zhang, X. Zhang, S. Wang, Near-optimal codes for information embedding in gray-scale signals, IEEE Transactions on Information Theory 56 (3) (2010) 1262–1270.

[42] A. D. Ker, R. Böhme, Revisiting weighted stego-image steganalysis, in: Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Vol. 6819, International Society for Optics and Photonics, 2008, p. 681905.

[43] J. Fridrich, M. Goljan, P. Lisonek, D. Soukal, Writing on wet paper, IEEE Transactions on signal processing 53 (10) (2005) 3923–3935.

[44] P. Bas, T. Filler, T. Pevnỳ, " break our steganographic system": the ins and outs of organizing boss, in: International workshop on information hiding, Springer, 2011, pp. 59–70.

[45] P. Bas, T. Furon., Bows-2. [online]., Available: http://bows2.ec-lille.fr (Jul. 2007).

[46] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[47] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images, IEEE Transactions on Information Forensics and Security 7 (3) (2012) 868–882.

[48] J. Kodovsky, J. Fridrich, V. Holub, Ensemble classifiers for steganalysis of digital media, IEEE Transactions on Information Forensics and Security 7 (2) (2012) 432–444. doi:10.1109/TIFS.2011.2175919.