IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 12, 2020, accepted October 20, 2020, date of publication October 26, 2020, date of current version November 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033814

A Preprocessing by Using Multiple Steganography for Intentional Image Downsampling on CNN-Based Steganalysis

HIROYA KATO, (Graduate Student Member, IEEE), KYOHEI OSUGE, SHUICHIRO HARUTA, (Member, IEEE), AND IWAO SASASE, (Senior Member, IEEE)
Department of Information and Computer Science, Faculty of Science and Technology, Keio University, Yokohama 223-8522, Japan

Corresponding author: Hiroya Kato (kato@sasase.ics.keio.ac.jp)

This work was supported in part by the Grant in Aid for Scientific Research from the Japan Society for Promotion of Science (JSPS) under Grant 17K06440.

ABSTRACT There exists a need of ''image steganalysis'' which reveals whether steganographic signals are embedded in an image to improve information security. Among various steganalysis, Convolutional Neural Networks (CNN) based steganalysis is promising since it can automatically learn the features of diverse steganographic algorithms. However, the detection performance of CNN is degraded when an image is intentionally resized by the nearest-neighbor interpolation before steganography. This is because spatial frequency in a resized image gets high, which disturbs the training. In order to overcome this shortcoming, in this article, we propose a preprocessing by using multiple steganography for intentional image downsampling on CNN-based steganalysis. In the proposed preprocessing, steganographic signals are additionally embedded into both resized original images and resized steganographic ones with the same embedding key since difference of spatial frequencies between them gets obvious, which helps CNN learn features. The reason why the difference gets obvious is that steganographic signals tend to be continuously embedded into same pixels in resized images when they are additionally embedded. Thus, by training resized images after the proposed preprocessing, the detection performance can be improved. Since the proposed preprocessing is very simple, it does not greatly increase the training time of CNN. Our evaluation shows accuracy in a model with the proposed preprocessing is up to 34.8% higher than that in the conventional model when the same steganography is additionally embedded. Besides, we also show that the proposed preprocessing yields up to 23.1% higher accuracy compared with the conventional one even when another steganography is additionally embedded.

INDEX TERMS Steganalysis, deep learning, image downsampling, convolutional neural networks.

I. INTRODUCTION

Image steganography is a technique which hides information in an image [1], [2], and it is concerned that attackers use it for malicious purposes [3]. For example, terrorists can use image steganography to share secret messages about the terrorism [4]. To deal with such risk, it is necessary to build the effective technique to reveal whether steganography is applied to an image, and it is called ''steganalysis''. The algorithms for steganography is called ''steganographic algorithms'', which are divided into ''spatial-domain steganography'' and ''JPEG steganography''. Spatial-domain steganography is applicable to almost all of the image formats such as PNG, BMP, and PGM. Steganographic signals are embedded into spatial domain in an image by altering the least significant bit or the low order 2 bits in some pixels, which are imperceptible by human eyes. On the other hand, JPEG steganography is applied to JPEG images. Steganographic signals are embedded by modifying quantized Discrete Cosine Transform (DCT) coefficients, which are components of JPEG images. In particular, we focus on spatial-domain steganography because it is applicable to more image formats. Thus, hereinafter, the word ''steganography'' means spatial-domain steganography. The image obtained after steganography is

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

called the ''**stego** image'', and the original image before steganography is called the ''**cover** image''. Moreover, the signal embedded by steganography is called ''stego signal''.

Existing steganalysis is mainly classified into the traditional Machine Learning (ML) based steganalysis [5], [6] and the Deep Learning (DL) based steganalysis [7]–[9]. In the traditional ML based steganalysis, handcrafted features based on the stego signals in images are trained. Finally, a trained binary classifier is used to detect whether stego signals are embedded into an input image. Although the traditional ML based steganalysis is useful for the targeted steganographic algorithms, designing effective features is a difficult task which needs strong knowledge of steganography. Thus, the steganalysis which does not depend on handcrafted features is desired. To cope with the limitation of the traditional ML based steganalysis, some researchers propose the DL based steganalysis [7]–[9] that can automatically learn effective features to detect stego images. In DL based steganalysis, a Convolutional Neural Network (CNN) is utilized since it is suitable for extracting features from images. By utilizing CNN, stego images obtained by various steganographic algorithms can be detected without handcrafted features. Although various steganalysis schemes have been proposed, we pay attention to [9] as the most useful conventional scheme because it is the state-of-the-art steganalysis that has versatility for various steganographic algorithms.

Meanwhile, recent steganographic algorithms such as WOW [10] and S-UNIWARD [11] tend to embed stego signals into noisy regions whose spatial frequencies are high with modifying pixel by ± 1 embedding. In noisy regions, the change of pixel values is drastic. Hence, stego signals embedded into noisy regions are hard to be detected since the change of pixel values by steganography is slight. In other words, a steganographic algorithm avoids embedding stego signals into smooth regions whose spatial frequencies are low. Because of this characteristic, a shortcoming on CNN-based steganalysis is concerned. That is, when an image is intentionally resized by the Nearest-Neighbor Interpolation (NNI) before steganography, the detection performance is degraded. In the case where an image is resized by the NNI, the changes of pixel values among neighboring pixels are sharp. As a result, the regions with high frequencies are distributed over a resized image, which means the spatial frequency of the resized image is high. This is because a pixel value of the original image itself is directly used as one of the resized image. In this case, since the correlations among neighboring pixels in resized images are originally complex, the slight embedding impact is hard to be noticeable. This is why the statistical modeling on resized images becomes more difficult when stego signals are embedded in such resized images. As a result, the useful features are hard to be obtained even by CNN, which incurs deterioration in detection performance. Thus, attackers can easily evade conventional CNN-based steganalysis because image downsampling is a simple way to make spatial frequencies of images high.

In order to overcome the shortcoming, in this article, we propose a preprocessing by using multiple steganography for intentional image downsampling on CNN-based steganalysis. In the proposed preprocessing, stego signals are additionally embedded into both resized cover images and resized stego ones with the same embedding key since the difference of spatial frequencies between them gets obvious, which helps CNN learn features. The reason why the difference gets obvious is that stego signals tend to be continuously embedded into the same pixels in resized images when they are additionally embedded. By training resized images after the proposed preprocessing, CNN can easily learn embedding patterns of stego signals in resized images, which can improve the detection performance. Thus, whenever resized images are trained in CNN or inspected whether they are stego ones or not, steganography is additionally applied to them by the proposed preprocessing. For example, in the case where steganography is additionally embedded into a stego image once, a stego image to which steganography is applied twice is obtained. Thus, in this case, an image is regarded as a stego one if the trained model judges steganography is applied to it twice; otherwise it is a cover one. Since the proposed preprocessing is very simple, it does not greatly increase the training time of CNN.

There exists work whose idea is similar to that of our work. In that work [12], its goal is to detect inconsistencies occurred during classification in image steganalysis by additionally embedding steganography. In other words, that work is a method that deal with the problem known as Cover Source Mismatch (CSM). In supervised machine learning based steganalysis, we must prepare a database of images for constructing training and testing dataset. When these datasets are created by images taken with a different conditions, various factors such as filters, zooms and compression algorithms are also different among the images. In this case, since the datasets has a bias problem, detection performance can be unreliable regardless of the approach of steganalysis. Thus, solutions to CSM are very important for practical steganalysis. On the other hand, our objective is to devise solutions for improving the detection performance of steganalysis for resized images. Thus, the most different point between two works is the problem to solve. In additional to the targeted problem, the contributions and results that we reveal in this work are considerably different from those of that work. In that work, the fact that additionally embedding is effective in improving detection performance for resized images is not demonstrated at all. Thus, our work is completely different from that work and useful in the domain of steganalysis. The contributions of this article are as follows:

1) We reveal the shortcoming that the NNI degrades the detection performance of a state-of-the-art CNN-based steganalysis even for known steganography.
2) We propose a preprocessing methodology overcoming the shortcoming by additionally embedding steganography. The proposed preprocessing is very simple and does not greatly increase training time of CNN.

3) To the best of our knowledge, this article is the first study which shows both the deterioration in the detection performance of CNN-based steganalysis because of image downsampling and countermeasures.

The rest of this article is constructed as follows. The attack model and related work are introduced in Section II. The shortcoming of the conventional CNN-based steganalysis is explained in Section III. The proposed methodology is described in Section IV. Various evaluation results and their interpretations are shown in Section V. Limitation and future work are explained in Section VI. Finally, the conclusions of this article are presented in Section VII.

II. ATTACK MODEL AND RELATED WORK

A. ATTACK MODEL

We assume terrorists try to share secret information about terrorism by using image steganography [4]. For instance, Terrorist A posts the image where information about the day of the terrorism is embedded on the imageboard. Terrorist B can understand secret information from the image by extracting stego signals in accordance with a certain rule. However, since the image just looks like a common image, it cannot be exposed as a stego image by human eyes. Besides, in the literature [3], [13], some researchers assume that a hacker exploits the stego image where malicious codes are embedded as stego signals to execute the malicious codes on a user's device downloading it. Thus, we also assume this type of attack.

B. RELATED WORK OF STEGANALYSIS

There exist a lot of schemes for steganalysis, which are roughly divided into ''Traditional ML based steganalysis'' and ''DL based steganalysis''. The representative schemes are explained in the next subsections.

1) TRADITIONAL MACHINE LEARNING BASED STEGANALYSIS

Traditional ML based steganalysis utilizes handcrafted features based on the stego signals extracted from images. In many cases, a binary classifier such as a support vector machine [14] or an ensemble classifier [15] is trained on the basis of extracted features to discriminate cover images from stego ones. In the practical scenario, the trained classifier is utilized to determine whether a input image is a cover one or a stego one.

Pevný *et al.* [5] propose a scheme leveraging the fact that the characteristics between adjacent pixels are different between natural images and stego ones. Authors focus on the fact that a standard method for embedding stego signals is the Least Significant Bit (LSB) replacement, in which LSBs of individual cover images are replaced with stego signals. That scheme utilizes higher order Markov chain for modeling the difference between adjacent pixels in natural images. That scheme can identify whether the difference is due to steganography in accordance with deviations from the model. However, that scheme is applicable only to simple LSB replacement. Recently, in order to make stego signals undetectable, various steganographic algorithms which minimize an impact of embedding and select regions for embedding in accordance with images are proposed.

To cope with various types of steganographic algorithms, Fridrich *et al.* propose a Spatial Rich Model (SRM) [6] which combines many diverse co-occurrence matrices to form a large feature vector. The authors focus on the feasibility that using submodels can consider various types of relationships among neighboring pixels of images. By using an ensemble classifier, that scheme can efficiently work with high-dimensional handcrafted features and large training datasets. Thus, the SRM can deal with various types of steganographic algorithms which are trained in the submodels. However, designing effective features is a difficult task which needs strong knowledge of steganography. Thus, steganalysis which does not depend on handcrafted features is desired.

2) DEEP LEARNING BASED STEGANALYSIS

To deal with the difficulties on traditional ML based steganalysis, many researchers propose steganalysis using DL that can automatically learn effective features to detect stego images. In general, a CNN is utilized since it is suitable for extracting features from images. Like other kinds of artificial neural networks, a CNN has an input layer, an output layer, and various hidden layers. Some of layers are convolutional ones using a mathematical model to pass on results to successive layers.

Given that the SRM [6] possesses a similar architecture to CNN, Tan and Li [7] propose the steganalysis using CNN. The authors expected that a well-trained CNN should be comparable to or even better than the performance of the SRM. However, experimental results show that the method is still inferior to the SRM since training time of CNN is too long to create well-trained model efficiently.

In order to facilitate efficient training of CNN, Qian *et al.* [8] propose a customized CNN-based model with a predefined high-pass filter in preprocessing layer. In general, the stego signals are very weak, which is greatly impacted by image content. Since a predefined high-pass filter can strengthen the weak stego signals and reduce the impact of cover images, the performance can be improved. Thus, compared to the SRM, that model achieves comparable performance on BOSSbase [16] and ImageNet [17] dataset without handcrafted features. However, a hand-designed filter such as a high-pass filter is detrimental to performance of CNN depending on the characteristics of steganographic algorithms. Hence, that model is inadequate to detect various steganographic algorithms effectively.

In order to be flexible for various types of steganographic algorithms, Boroumand *et al.* [9] propose a Steganalysis Residual Network (SRNet) based model designed to minimize the use of heuristics and externally enforced elements. That model includes no fixed preprocessing layers such as

a high-pass filter. This is because the network can automatically learn the best filters for convolution via end-to-end training. The key part of that model is to involve the so-called residual shortcuts that have been shown in the literature [18] to help learn the parameters in upper layers of deep networks. Besides, average pooling of front segment is not incorporated to prevent suppression of the stego signals. By these techniques, that model can flexibly learn stego signals and be applied to various steganographic algorithms without heuristics and externally enforced elements. Although various steganalysis schemes have been proposed, we mainly utilize [9] in the experiments after this section because it is the state-of-the-art steganalysis model that has versatility for various steganographic algorithms.

C. RELATED WORK OF STEGANOGRAPHY

In general, embedding stego signals affects the quality of images. Thus, steganographic algorithms select pixels and regions for embedding to minimize an impact of embedding. As a representative steganography, Filler and Fridrich [19] propose a steganography called HUGO-BD. In particular, HUGO-BD tends to embed stego signals into edges that are smooth along any direction in images. Embedding stego signals into edges is effective to minimize an impact of embedding because the edges are originally located in noisy regions in images. However, stego signals along an edge can be easily detected by existing steganalysis using features based on distortion in every direction. In order to realize more undetectable embedding, recent representative steganographic algorithms such as WOW [10] and S-UNIWARD [11] tend to carefully embed stego signals into noisy regions so as not to create edges after steganography. Thus, this clever embedding can make it difficult to judge whether high frequencies are caused by steganography or not.

III. SHORTCOMING OF CONVENTIONAL CNN-BASED STEGANALYSIS MODEL

A. OVERVIEW OF THE SHORTCOMING

We assume that the detection performance on CNN-based steganalysis is degraded when an image is resized by the NNI before steganography. In the steganography used in this work, stego signals are embedded by modifying pixel values in cover images by ± 1 change. In this case, compared with the cover images, the regions in which the changes of pixel values among neighboring pixels become unnaturally sharp appear in stego images because of an impact of embedding. This means that the spatial frequencies of stego images are slightly higher than that of cover ones. In other words, steganography slightly increases the pixels whose pixel values sharply change among neighboring pixels. Thus, in order to minimize this impact, steganographic algorithms select pixels and regions for embedding. In particular, stego signals tend to be embedded into noisy regions whose spatial frequencies are high because the slight changes of pixel values in such regions are not perceptible by human eyes. Such embedding also makes it difficult for CNN-based steganalysis to detect the signals. CNN-based steganalysis can find presence of stego signals on the basis of slight differences of statistical features between cover and stego images. The statistical features are obtained from the correlations among neighboring pixels. Because the correlations in noisy regions are complicated, statistical modeling is not easy even for CNN. In other words, CNN must find slight changes by steganography from the complicated correlations among pixels. In normal images without resizing, since regions with high frequencies are limited, there exist regions in which embedding impacts after steganography are noticeable. Hence, this fact results in obvious differences of spatial frequencies between cover and stego images. Since statistical modeling for such images is relatively easy, the features about steganography can be successfully obtained after long training process of CNN.

However, when an image is resized by the NNI, the changes of pixel values among neighboring pixels are extremely sharp. This is because a pixel value of the original image itself is directly used as one of the resized image. As a result, the regions with high frequencies are distributed over a resized image, which means spatial frequency of the resized image is originally high. Since correlations among neighboring pixels in resized images are originally complex, the slight embedding impacts are hard to be noticeable. In this case, the differences of spatial frequencies between cover and stego images are small in comparison to the differences in normal images. This is why the statistical modeling becomes more difficult when stego signals are embedded in resized images. Accordingly, the useful features are hard to be obtained even by CNN, which incurs deterioration in detection performance. Thus, this simple operation makes it difficult for CNN-based steganalysis to detect steganography.

B. VALIDATION OF THE SHORTCOMING

In order to validate the degrading detection performance due to the NNI, we evaluate the accuracy of the conventional model [9] for the resized images and the cropped ones. In this validation, we embed WOW at the embedding payload 0.4 bits-per-pixel (bpp). Fig. 1 shows how to create the resized dataset and the cropped one. By using the NNI, we resized the 10,000 original images from their original size 512×512 to 256×256 to obtain Resized Cover Images (hereinafter, they are called RCI). Furthermore, we cropped original images and only utilized the upper-left images of them as Cropped Cover Images (hereinafter, they are called CCI). This is because CNN is sensitive to the number of the data. In other words, cropping is used in order to match the size of original images to that of resized images without downsampling. The results may depend on where the images are cropped. However, these depend on the distribution of noisy regions in original images. We use BOSSbase dataset ver 1.01 [16] as an image dataset. Fig. 2 shows sample images in BOSSbase dataset ver 1.01. As we can see from Fig. 2, the difference between the upper-left region and the central one of each image is dependent on an original image.

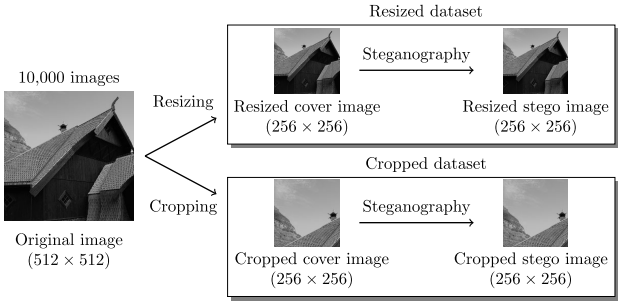


FIGURE 1. How to create the resized dataset and the cropped one.

FIGURE 2. Sample images in BOSSbase ver 1.01.

This is why the location of images for cropping does not necessarily impact the detection performance. Since cropped images can be regarded as new images, they are used to increase the number of dataset in related work [20]. Thus, we only use the upper-left images in this experiment. After obtaining two types of cover images, by embedding steganography into CCI and RCI, we obtained 10,000 Cropped Stego Images (hereinafter, they are called CSI_1) and 10,000 Resized Stego Images (hereinafter, they are called RSI_1), respectively. Finally, the 5,000 RCI and their 5,000 RSI_1 are trained by the CNN, and the others are tested. Similarly, the 10,000 CCI and the 10,000 CSI_1 are also evaluated by the same strategy. The simulation parameters used in this simulation are the same ones described in Section V-A.

Fig. 3 shows accuracy in SRNet [9] for the resized dataset and the cropped one. As shown in Fig. 3, the training for the resized dataset does not progress well until around 10,000 iteration whereas that for the cropped dataset progresses well. This is because high frequencies of resized images disturb efficient training of CNN. Since WOW tends to embed stego signals into noisy regions, there exist few signals isolated in smooth regions. Thus, CNN should train features by focusing on a slight difference in noisy regions between resized cover images and resized stego ones. However, by the NNI, a spatial frequency of a resized image inevitably gets quite high even if that of an original image is low, and noisy regions are increased in RCI. Thus, although spatial frequencies of RSI_1 get slightly high by steganography, the difference of frequencies between the RCI and RSI_1 is very small. In this case, because the training is more difficult, CNN cannot learn features until it repeats training to some extent. As a result, the detection performance of CNN-based steganalysis is degraded because of the unclear difference of frequency. In fact, the testing accuracy is up to 78.6% in the resized dataset whereas that in the cropped one

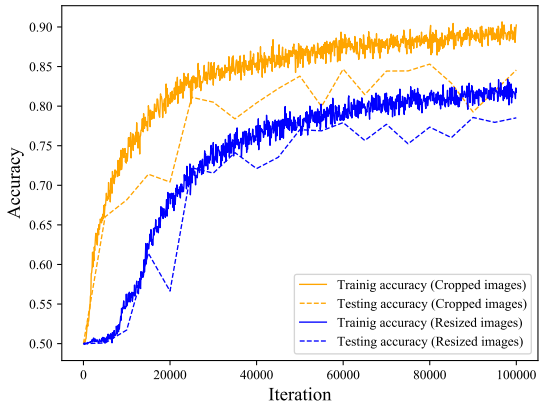


FIGURE 3. Accuracy in SRNet for the resized dataset and the cropped one.

is up to 85.3%. Thus, we conclude that the requirement to improve detection performance is to enlarge the difference of frequency between RCI and RSI_1.

This result demonstrates that the shortcoming is valid. Because attackers only have to use the NNI, they can easily evade the state-of-the-art CNN-based steganalysis. Although there has been a study regarding the effects of image downsampling on traditional ML based steganalysis [21], to the best of our knowledge, there has still been no studies which investigate both the effect on CNN-based steganalysis and countermeasures. Hence, our goal is to design a countermeasure against resized stego images created by the NNI.

IV. PROPOSED METHODOLOGY

In order to achieve our goal, we propose a preprocessing by using multiple steganography for intentional image downsampling on CNN-based steganalysis. Through many inspections and experiments, we found the phenomenon that recent steganographic algorithms tend to embed most stego signals into the same pixels in resized images when they are additionally embedded with the same embedding key. In the case where steganography is embedded into resized images twice, spatial frequencies of the Resized Stego Images to which steganography is embedded twice (hereinafter, these are called RSI_2) get further high, and difference of frequencies between RSI_1 and RSI_2 is enlarged. Thus, we assumed that by leveraging this phenomenon, the requirement mentioned in Section III-B are met, and detection performance is improved. We dare to apply steganography to RCI and RSI_1 n times as preprocessing. The value of n is different depending on embedding payloads of stego signals embedded in images of a training dataset. This is because the degree of enlarging the difference of spatial frequencies between RCI and RSI_1 per one embedding is different for each payload. Thus, we obtain RSI_n and RSI_{n+1} which denote resized stego images after n additional embeddings for RCI and RSI_1, respectively. Besides, we make CNN train RSI_n and RSI_{n+1} instead of RCI and RSI_1, respectively. Thus, when an image is regarded as RSI_{n+1}, it turns out that it is a stego one; otherwise it is a cover one. Since the proposed preprocessing

is very simple, it does not greatly increase the training time of CNN. In the following subsections, we first validate our assumption and explain the reason why the difference of the frequency is enlarged by additionally embedding. Finally, the proposed preprocessing methodology is described.

A. VALIDATION OF OUR ASSUMPTION

In order to validate our assumption, we conducted simple inspections in the case where stego signals are additionally embedded up to twice. We show that the difference of frequency between RSI_1 and RSI_2 is larger than that between RCI and RSI_1 by using numerical value. In the following subsection, we firstly describe how to numerically express a frequency of each pixel and an image. After that description, we show the comparison of RCI, RSI_1, and RSI_2 with regard to the frequencies of them.

1) NUMERICAL EXPRESSION OF FREQUENCY IN AN IMAGE

In order to numerically express a frequency of each pixel and an image, we define two High Frequency Degree (HFD) for them. For each pixel, we define the absolute values resulting from convolving the high-pass filter with pixel values around a targeted pixel as $\mathrm{HFD}_{\mathrm{pixel}}$. This is because the higher a spatial frequency around a pixel is, the larger the absolute value is. In addition to that, we define the average of $\mathrm{HFD}_{\mathrm{pixel}}$ in an image as $\mathrm{HFD}_{\mathrm{image}}$. We select the following filter

$$HPF = \begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix}, \quad (1)$$

as a high-pass filter, where HPF denotes the high-pass filter used for calculating $\mathrm{HFD}_{\mathrm{pixel}}$ and $\mathrm{HFD}_{\mathrm{image}}$. This is because it is commonly utilized in the domain of steganalysis [6], [8], [22]. $\mathrm{HFD}_{\mathrm{pixel}}$ of $p(x, y)$ which denotes a pixel value is calculated as

$$H_{\mathrm{pixel}}(x, y) = \left| \sum_{i=0}^{4} \sum_{j=0}^{4} h(i, j) p(x + i - 2, y + j - 2) \right|, \quad (2)$$

where $h(i, j)$ denotes an element of HPF of (1). Let $\mathrm{I}_{\mathrm{input}}$ denote a $n \times m$ input image. $H_{\mathrm{image}}(\mathrm{I}_{\mathrm{input}})$ which denotes $\mathrm{HFD}_{\mathrm{image}}$ of $\mathrm{I}_{\mathrm{input}}$ is calculated as

$$H_{\mathrm{image}}(\mathrm{I}_{\mathrm{input}}) = \frac{1}{nm} \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} H_{\mathrm{pixel}}(x, y). \quad (3)$$

By using $\mathrm{HFD}_{\mathrm{pixel}}$ and $\mathrm{HFD}_{\mathrm{image}}$, we can numerically express a spatial frequency of each pixel and an image.

2) COMPARISON OF $\mathrm{HFD}_{\mathrm{image}}$

In order to confirm that the difference of $\mathrm{HFD}_{\mathrm{image}}$ gets remarkable by additionally embedding steganography into resized images, we inspected the averages of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when steganography is embedded up to twice. Table 1 shows the averages of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when WOW is embedded at 0.4 bits-per-pixel (bpp) up to twice. Note that we create these stego images by embedding stego signals with the same key. As shown in Table 1, the difference of the average of $\mathrm{HFD}_{\mathrm{image}}$ between RSI_1 and RSI_2 is about 2.1 times larger than that between RCI and RSI_1. Meanwhile, Table 2 shows the averages of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when WOW is embedded at 0.2 bpp up to twice. As shown in Table 2, the difference is enlarged although it is smaller than that in the case of 0.4 bpp. Thus, multiple embeddings are required so as to enlarge the difference of $\mathrm{HFD}_{\mathrm{image}}$ as large as that in the case of 0.4 bpp. Besides, Table 3 shows the average $\mathrm{HFD}_{\mathrm{image}}$ when S-UNIWARD is embedded at 0.4 bpp. As shown in Table 3, there also exists the similar tendency in embedding S-UNIWARD.

Meanwhile, Table 4 shows the average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 cropped images when WOW is embedded at 0.4 bpp up to twice. As shown in Table 4, the difference of $\mathrm{HFD}_{\mathrm{image}}$ between CSI_1 and Cropped Stego Images which steganography is embedded twice (hereinafter, these are called CSI_2) is also larger than that between CCI and CSI_1. These results mean that additionally embedding is effective in enlarging $\mathrm{HFD}_{\mathrm{image}}$ even in resized images although the degree is small compared with $\mathrm{HFD}_{\mathrm{image}}$ in cropped images. Furthermore, compared with Table 3, the average of $\mathrm{HFD}_{\mathrm{image}}$ in RCI is about 3.1 times larger than that of $\mathrm{HFD}_{\mathrm{image}}$ in CCI, and it means that the spatial frequencies of resized images are considerably high.

TABLE 1. The average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when WOW is embedded at 0.4 bpp up to twice.

	RCI	RSI_1	RSI_2
Average of $\mathrm{HFD}_{\mathrm{image}}$	99.501	99.786	100.389
Difference from the left value	–	0.285	0.603

TABLE 2. The average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when WOW is embedded at 0.2 bpp up to twice.

	RCI	RSI_1	RSI_2
Average of $\mathrm{HFD}_{\mathrm{image}}$	99.501	99.591	99.805
Difference from the left value	–	0.090	0.214

TABLE 3. The average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images when S-UNIWARD is embedded at 0.4 bpp up to twice.

	RCI	RSI_1	RSI_2
Average of $\mathrm{HFD}_{\mathrm{image}}$	99.501	99.753	100.375
Difference from the left value	–	0.252	0.622

TABLE 4. The average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 cropped images when WOW is embedded at 0.4 up to twice.

	CCI	CSI_1	CSI_2
Average of $\mathrm{HFD}_{\mathrm{image}}$	32.214	32.865	34.145
Difference from the left value	–	0.651	1.28

TABLE 5. The average ratio of pixels changed in 10,000 cropped images and 10,000 resized ones after WOW is embedded at 0.4 bpp up to twice. (The unit is %).

Image Type	Returned pixels	+1	−1	+2	−2
Resized	2.30	2.83	0.0292	46.5	48.3
Cropped	5.57	6.16	0.135	41.6	46.6

From these results, our assumption is valid, and we conclude that enlarging the difference of frequency is possible by additionally embedding. Therefore, we expect that detection performance is improved by training RSI_n and RSI_{n+1} instead of RCI and RSI_1, respectively.

B. ENLARGING DIFFERENCE OF FREQUENCY IN RESIZED IMAGES

1) PIXEL MODIFICATION BY ADDITIONALLY EMBEDDING

When additionally embedding steganography into RCI, the spatial frequencies of RSI_2 inevitably get higher than that of RSI_1. This is because most embedding changes tend to continuously occur at the same pixels by additionally embedding with the same embedding key. Table 5 shows the average ratio of pixels changed in 10,000 cropped images and 10,000 resized ones after WOW is embedded at 0.4 bpp up to twice. Because stego signals are embedded twice, pixel values are moved to up to ±2 away from original values in cover images. Furthermore, when the signals are successively embedded in the same pixels in different directions during two embeddings, there exist pixels whose values return to original values. As we can see from Table 5, more than 90% of pixel values are moved to ±2 away from original values, which means stego signals tend to be successively embedded in the same pixels regardless of image type. This is because stego signals are embedded with the same key. The embedding directions (±1) for each pixel are decided depending on a random sequence based on a used key. When the same key is used, the embedding simulator always generates the same random sequence. In this case, unless the embedding costs of pixels are largely changed during two embeddings, not only targeted pixels but also embedding directions should be the same ones. However, there exists the difference of the tendencies between cropped and resized images. In comparison with the ratio of ±1 in resized images, that in cropped ones is high. In particular, as for the ratio of +1, that of cropped images is 3.33% higher than that of resized ones. The reason why such difference appears is that there exist more pixels with high frequencies in resized images. Fig. 4 shows the distribution of $\mathrm{HFD}_{\mathrm{pixel}}$ in a cropped image and a resized one. As shown in Fig. 4, the distribution of $\mathrm{HFD}_{\mathrm{pixel}}$ in a resized image is widely spread compared with a cropped one. Thus, when the first steganography is applied to RCI, most stego signals are embedded into pixels whose $\mathrm{HFD}_{\mathrm{pixel}}$ are high. After the first embedding, although the distribution in RSI_1 is changed, it is the slight change. Thus, since the cost of pixels for embedding stego signals is hardly changed, the second steganography also embeds most stego signals into the same pixels in RSI_1. Besides, the directions of embedding are also similar to the first embedding.

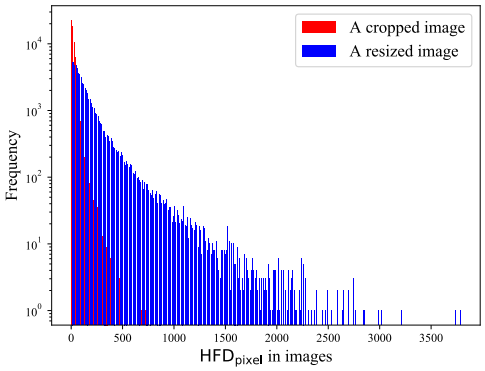


FIGURE 4. The distribution of $\mathrm{HFD}_{\mathrm{pixel}}$ in a cropped image and a resized one.

On the other hand, in CCI, the distribution is narrowly spread. Because of the narrow distribution, the embedding costs of pixels are easy to be changed during two embeddings. Thus, stego signals tend to be embedded into different pixels in cropped images compared with resized images.

2) EFFECT OF ADDITIONALLY EMBEDDING

Let $p_{n'}(x, y)$ denote a pixel value in a stego image after steganography is applied n' times. When $n' \geq 1$, $p_{n'}(x, y)$ is defined as

$$p_{n'}(x, y) = p_{n'-1}(x, y) + e_{n'}(x, y), \tag{4}$$

where $e_{n'}(x, y) \in \{-1, 0, +1\}$ mean embedding changes by the n'-th embedding. Let $p^{\mathrm{next}}_{n'}(x, y)$ denote a pixel value next to $p_{n'}(x, y)$. Equation (4) is also applicable to $p^{\mathrm{next}}_{n'}(x, y)$. The difference $dp_{n'}(x, y)$ between $p_{n'}(x, y)$ and $p^{\mathrm{next}}_{n'}(x, y)$ is expressed as

$$dp_{n'}(x, y) = p_{n'}(x, y) - p^{\mathrm{next}}_{n'}(x, y). \tag{5}$$

According to (4) and (5), $dp_{n'}(x, y)$ is represented as

$$\begin{aligned} dp_{n'}(x, y) &= p_{n'}(x, y) - p^{\mathrm{next}}_{n'}(x, y), \\ &= p_{n'-1}(x, y) + e_{n'}(x, y) \\ &\quad - p^{\mathrm{next}}_{n'-1}(x, y) - e^{\mathrm{next}}_{n'}(x, y), \\ &= dp_{n'-1}(x, y) + e_{n'}(x, y) - e^{\mathrm{next}}_{n'}(x, y), \\ &= dp_{n'-1}(x, y) + \Delta e_{n'}, \end{aligned} \tag{6}$$

where $\Delta e_{n'}$ means the changes of pixel values by the n'-th embedding. When both $p_{n'-1}(x, y)$ and $p^{\mathrm{next}}_{n'-1}(x, y)$ are modified by ±1, and $p_{n'-1}(x, y) \geq p^{\mathrm{next}}_{n'-1}(x, y)$, $\Delta e_{n'}$ is represented as

$$\Delta e_{n'} = \begin{cases} 2, & \text{if } e_{n'}(x, y) = 1 \text{ and } e^{\mathrm{next}}_{n'}(x, y) = -1, \\ -2, & \text{if } e_{n'}(x, y) = -1 \text{ and } e^{\mathrm{next}}_{n'}(x, y) = 1, \\ 0, & \text{if } e_{n'}(x, y) = e^{\mathrm{next}}_{n'}(x, y). \end{cases} \tag{7}$$

When only $p_{n'-1}(x, y)$ is modified by ± 1, and $p_{n'-1}(x, y) \geq p^{\text{next}}_{n'-1}(x, y)$, $\Delta e_{n'}$ is expressed as

$$\Delta e_{n'} = \begin{cases} 1, & \text{if } e_{n'}(x, y) = 1 \text{ and } e^{\text{next}}_{n'}(x, y) = 0, \\ -1, & \text{if } e_{n'}(x, y) = -1 \text{ and } e^{\text{next}}_{n'}(x, y) = 0. \end{cases} \tag{8}$$

In this work, the same key is used for every embedding. This is why most stego signals tend to be repeatedly embedded in the same pixels when steganography is applied to the same image multiple times. Furthermore, embedding directions (± 1) for targeted pixels also tend to be the same as the directions of the previous embedding unless embedding costs of them are not greatly changed. As signals are additionally embedded in the same pixels to yield $dp_{n'}(x, y) = dp_{n'-1}(x, y)+2$ and $dp_{n'}(x, y) = dp_{n'-1}(x, y)+1$, the changes of the pixel values around the pixels become sharp compared with the previous state. Because stego signals are originally embedded in stego images once, a cover image is always equivalent to the previous state of a stego image even after additional embeddings. In other words, the difference of pixel values in cover images and that in stego images are equivalent to $dp_{n'-1}(x, y)$ and $dp_{n'}(x, y)$, respectively. As a result, unnatural regions in stego images appear prior to the appearance in cover images by additional embeddings. This helps CNN learn the impacts of embedding stego signals.

Furthermore,, after both $p_{n'-2}(x, y)$ and $p^{\text{next}}_{n'-2}(x, y)$ are modified in the same direction by the $n' - 1$-th embedding, changing only embedding directions for $p_{n'-1}(x, y)$ by the n'-th embedding makes the embedding impact more noticeable. For example, in the case where steganography is applied to an image up to twice, namely $n' = 2$, $p_{n'-2}(x, y) > p^{\text{next}}_{n'-2}(x, y)$ means $p_{\text{cover}}(x, y) > p^{\text{next}}_{\text{cover}}(x, y)$. $p_{\text{cover}}(x, y)$ and $p^{\text{next}}_{\text{cover}}(x, y)$ mean a pixel value in a cover image and a pixel value next to $p_{\text{cover}}(x, y)$, respectively. According to (6) and (7), when the first embedding occurs to meet $e_1(x, y) = e^{\text{next}}_1(x, y)$, the following equation is established

$$dp_1(x, y) = dp_{\text{cover}}(x, y), \tag{9}$$

where $dp_{\text{cover}}(x, y)$ means the difference between $p_{\text{cover}}(x, y)$ and $p^{\text{next}}_{\text{cover}}(x, y)$. This means that the difference of pixel values between a cover and a stego image is not changed by the first embedding although signals are embedded in both $p_{\text{cover}}(x, y)$ and $p^{\text{next}}_{\text{cover}}(x, y)$. After that, when the second embedding changes $p_1(x, y)$ and $p^{\text{next}}_1(x, y)$ to meet $e_2(x, y) = 1$ and $e^{\text{next}}_2(x, y) = -1$, according to (6) and (7), the following equation is established

$$dp_2(x, y) = dp_1(x, y) + 2. \tag{10}$$

In this case, the additional embedding makes the embedding impact more clear. Consequently, when CNN compares $dp_2(x, y)$ with $dp_1(x, y)$ instead of comparing $dp_1(x, y)$ with $dp_{\text{cover}}(x, y)$, the presence of the stego signal gets more distinguishable. These phenomenons occur in various pixels within an image, which achieves improvement of detection performance. In the above case, $H_{\text{pixel}}(x, y)$ of $p_{n'}(x, y)$ is larger than $H_{\text{pixel}}(x, y)$ of $p_{n'-1}(x, y)$ due to definition in (1) and (2). In this case, since the spatial frequency greatly increases, the differences of the spatial frequencies between cover and stego images are enlarged, which helps CNN train useful evidence of the presence of stego signals.

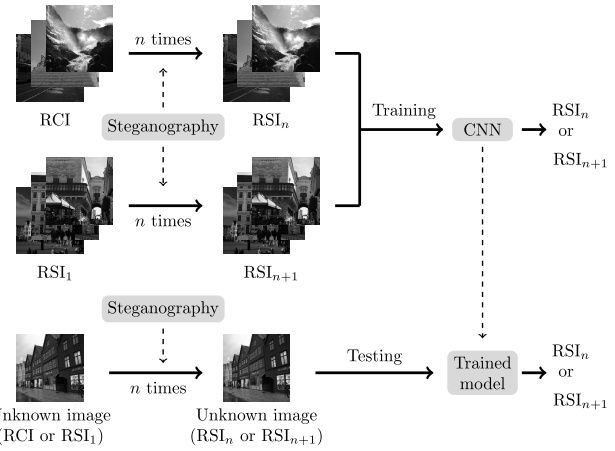


FIGURE 5. The proposed preprocessing methodology.

C. PREPROCESSING METHODOLOGY FOR RESIZED IMAGES

In this section, we explain the proposed preprocessing methodology in detail. Fig.5 shows the proposed preprocessing methodology. As shown in Fig.5, the CNN trains RSI$_n$ and RSI$_{n+1}$ instead of RCI and RSI$_1$, respectively. In the testing phase, we dare to embed steganography into unknown images n times before they are tested by the trained model. When unknown images are regarded as RSI$_{n+1}$, it turns out that they are RSI$_1$; otherwise they are RCI. The value of n is different depending on each embedding payload of stego signals embedded in images of training datasets. In this methodology, it is assume that unknown images are resized ones because our scope is steganalysis for resized images. In the following sections, we show the algorithm to describe the proposed preprocessing step by step.

1) PREPROCESSING ON TRAINING PHASE

The training dataset of resized cover images is represented as

$$C_{\text{train}} = \{c_i | 1 \leq i \leq m_{\text{cover}}\}, \tag{11}$$

where m_{cover} is the total number of cover images. Furthermore, the training dataset of resized stego images created from $c_i \in C_{\text{train}}$ is represented as

$$S^{(sa,p,k)}_{\text{train}} = \{s^{sa,p,k}_i | 1 \leq i \leq m_{\text{cover}}\}, \tag{12}$$

where sa, p, and k denote embedded steganographic algorithm, embedding payload, and a embedding key, respectively. The number of resized stego images is equal to that of resized cover ones since c_i and $s^{sa,p,k}_i$ are paired images.

Additional embedding with sa at p bpp is carried out for $s^{sa,p,k}_i \in S^{(sa,p,k)}_{\text{train}}$. Let $n_{(sa,p,k)}$ denote the number of additional embeddings for $S^{(sa,p,k)}_{\text{train}}$. $n_{(sa,p,k)}$ is determined by preliminary experiments. The experiments reveals how many times

the additional embeddings are required to improve detection performance on $S^{(sa,p,k)}_{\text{train}}$.

Thus, $n_{(sa,p,k)}$ is changed depending on sa, p, and k. The same k used in (12) is utilized for the additional embeddings. The training dataset of resized stego images after the $n_{(sa,p,k)}$-th additional embedding is represented as

$$S^{n^{(sa,p,k)}}_{\text{train}} = \{s_i^{n^{sa,p,k}} | 1 \leq i \leq m_{\text{cover}}\}. \tag{13}$$

Similarly, additionally embedding is also conduced for $c_i \in C_{\text{train}}$. The training dataset of resized cover images after the $n_{(sa,p,k)}$-th additional embedding is expressed as

$$C^{n^{(sa,p,k)}}_{\text{train}} = \{c_i^{n^{sa,p,k}} | 1 \leq i \leq m_{\text{cover}}\}. \tag{14}$$

By using pairs of $c_i^{n^{sa,p,k}} \in C^{n^{(sa,p,k)}}_{\text{train}}$ and $s_i^{n^{sa,p,k}} \in S^{n^{(sa,p,k)}}_{\text{train}}$ for the training of CNN, a trained model $M_{n_{(sa,p,k)}}$ are created. Finally, $M_{n_{(sa,p,k)}}$ is utilized to identify resized stego images in the testing phase.

2) PREPROCESSING ON TESTING PHASE

In the testing phase, an unknown image without a label is judged whether it is a stego image or not. A set of unknown images is represented as

$$U = \{u_j | 1 \leq j \leq m_{\text{unknown}}\}, \tag{15}$$

where m_{unknown} is the total number of unknown images. Depending on the prepared $M_{n_{(sa,p,k)}}$, $n_{(sa,p,k)}$ additional embeddings are applied to $u_j \in U$. After that, a label of $u_j \in U$ is predicted by $M_{n_{(sa,p,k)}}$.

V. EVALUATION

In order to demonstrate the effectiveness of the proposed preprocessing, we evaluate Accuracy (ACC) in various situations with real image datasets. ACC is calculated as

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{16}$$

where TP, TN, FP, and FN denote the number of True Positive (stego images are regarded as stego ones), True Negative (cover images are regarded as cover ones), False Positive (cover images are regarded as stego ones), and False Negative (stego images are regarded as cover ones), respectively.

A. SIMULATION PARAMETERS

Table 6 shows our simulation parameters. We use the image dataset called BOSSbase dataset ver. 1.01 [16] for our experiments and evaluations. We obtained this dataset from Binghamton website [24] to conduct our work. It contains 10,000 uncompressed gray scale images including landscape, buildings, and animals with the size of 512 × 512. The images are taken with eight different cameras. This is why our experimental results directly demonstrate that the proposed preprocessing is applicable even to an image actually collected by a camera. The BOSSbase is a famous dataset in the domain of steganalysis and steganography because it is often utilized for experiments and evaluations. In order

TABLE 6. Simulation parameters.

Name of parameter		Value
Image dataset	Source	BOSSbase [16]
	Number	10,000
	Size	512 × 512
	File extension	.pgm
Steganography		WOW [10] S-UNIWARD [11] HUGO-BD [19]
Number of training data		10,000 (5,000 cover images and their stego ones)
Number of testing data		10,000 (5,000 cover images and their stego ones)
Bacth size	Training	20
	Testing	40
Iteration		100,000
Used CNN-based steganalysis		SRNet [9] and Xu-Net [23]
Used feature-based steganalysis		SRM [6]
Simulation tool		Python (TensorFlow)

to show the improvement of the detection performance in resized images, the images are resized to 256 × 256 images by the NNI as shown in Fig. 1. We randomly select 5,000 RCI from the resized images and their RSI_1 are utilized for training on CNN model. The other 5,000 RCI and their RSI_1 are used for testing. We use the C++ implementations of steganographic algorithms which are downloaded from the website [24] in order to embed stego signals with WOW [10], S-UNIWARD [11], and HUGO-BD [19] in our evaluation. Stego signals are embedded at 0.2 and 0.4 bits-per-pixel (bpp) because they are the standard embedding payload which is used to evaluate performance in various literatures [6], [8], [9], [25], [26]. We use the same embedding key for different images and different embedding times by the proposed preprocessing. We decided how to generate stego images after preliminary experiments described in Section V-B. We use an optimal simulator for generating stego images. The batch size in training, that in testing and iteration are 20, 40, and 100,000, respectively. In CNN, the batch size is the number of samples in one training process or one testing process. One iteration means one batch is processed. Thus, the CNN trains 100,000 times with the batch size of 20 in our simulations. We utilize SRNet [9] as the state-of-the-art conventional model for the evaluation. Moreover, we also use Xu-Net [23] and Spatial Rich Model (SRM) [6] as traditional schemes in our evaluation. Xu-Net is a representative CNN-based steganalysis, which is known as fundamental CNN model for steganalysis. In terms of SRM, many handcrafted features for modeling noise component of images are extracted from images. In our evaluation, an ensemble classifier [27] is used for training SRM features and predicting the labels of input images. SRNet and Xu-Net are implemented by TensorFlow in Python. Moreover, we use the MATLAB implementation of SRM obtained from the Binghamton website [24]. We create Prop. (SRNet), Prop. (Xu-Net), and Prop. (SRM), namely the proposed models made by incorporating the proposed preprocessing with each conventional steganalysis and compare these models. Thus, the proposed model trains resized cover images and resized stego ones after additionally embedding by the proposed preprocessing.

B. PRELIMINARY EXPERIMENTS OF EMBEDDING KEY

We use the same embedding key for different images when creating stego images. Furthermore, the same key is used for different embedding times. We decided to generate stego images with these strategies after preliminary experiments regarding the correlation between the detection performance and patterns of used keys. Table 7 shows the max testing ACC of SRNet and Prop. (SRNet) for resized images in several cases of used keys. In this experiment, WOW is embedded at 0.4 bpp. The term ''First embedding'' in Table 7 means the embedding for creating stego images from cover ones. Although the embedding is generally conducted once, our work carries out multiple embeddings. Thus, we dare to call the usual embedding for cover images ''first embedding''. In terms of Prop. (SRNet), there exist two schemes depending on the keys used for additional embeddings of the proposed preprocessing. As shown in Table 7, the best testing ACC is achieved when the same key is used for both different images and different embedding times. When additionally embedding with different keys is carried out for the case where the first embedding is conducted with the same key, the testing ACC is 0.5, which means additionally embedding is not effective. In this case, because stego signals are not embedded in the same pixels, such embedding makes the statistical features of the signals more confused.

TABLE 7. The max testing ACC of SRNet and Prop. (SRNet) for resized images in several cases of used key. (WOW is embedded at 0.4 bpp).

	First embedding	
	Same key	Different key
Prop. (SRNet) with same key	0.856	0.500
Prop. (SRNet) with different key	0.500	0.500
SRNet	0.786	0.500

On the other hand, when the keys for the first embedding are different among images, the detection performance is not improved regardless of cases of keys used by additional embedding. From these results, we conclude that a key used for the additional embedding must be the same as the key used for the first embedding to improve the detection performance. This situation means that attackers create stego images with the same key, which is a simple situation. However, to the best of our knowledge, there is no useful study and effective solution to resized stego images even in this simple situation. To devise the practical solutions, we must deal with the simple situation in the first place. Therefore, in the following experiments in this work, we use the datasets created with the same key.

C. COMPARISON WITH STATE-OF-THE-ART MODEL

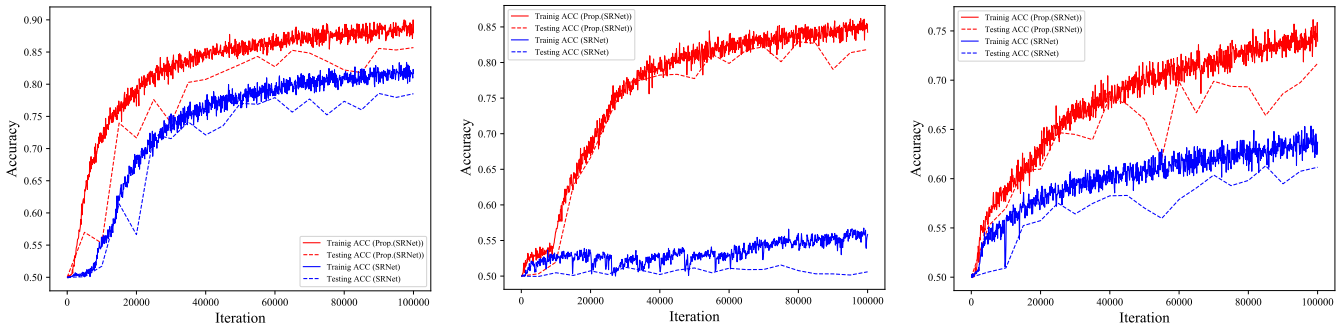
In order to show the effectiveness of the proposed preprocessing for resized images, we compare the Prop. (SRNet) with SRNet. In this evaluation, the proposed preprocessing embeds the same steganography as that embedded in RSI_1 at 0.4 bpp once. Thus, we obtain RSI_1 and RSI_2 by additionally applying steganography to RCI and RSI_1, respectively.

TABLE 8. The max testing ACC of the Prop. (SRNet) and SRNet for resized datasets. Payload is 0.4 bpp.

Model	WOW	S-UNIWARD	HUGO-BD
Prop. (SRNet)	**0.856**	**0.829**	**0.717**
SRNet	0.786	0.516	0.613

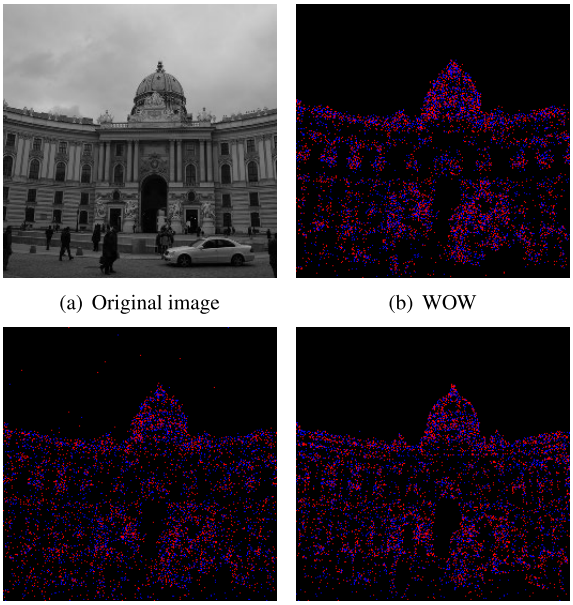
In SRNet, CNN simply trains the RCI and RSI_1. In the Prop. (SRNet), CNN trains RSI_1 and RSI_2 instead of RCI and RSI_1, respectively. Table 8 shows the max testing ACC of the Prop. (SRNet) and SRNet for resized dataset. As shown in Table 8, the Prop. (SRNet) improves detection performance for three steganographic algorithms compared with SRNet. This is because the difference of the average of HFD_{image} between RSI_1 and RSI_2 gets larger than that between RCI and RSI_1 by the proposed preprocessing, which help CNN learn embedding patterns. In particular, with regard to S-UNIWARD, the testing ACC in the SRNet, is up to 51.6%, whereas the Prop. (SRNet) achieves the testing ACC of 82.9%, 31.3% higher than that of SRNet.

Furthermore, Fig. 6 shows the progress regarding ACC of SRNet and the Prop. (SRNet) in the case where the same steganography is additionally embedded at 0.4 bpp once. As shown in Fig. 6(a), the training of the Prop. (SRNet) for WOW progresses well whereas SRNet cannot efficiently learn until around 10,000 iteration. Similarly, in the case of HUGO-BD, the Prop. (SRNet) facilitates training of CNN as shown in Fig. 6(c). In terms of S-UNIWARD, as shown in Fig. 6(b), although the training of the Prop. (SRNet) does not progress well until around 10,000 iteration, it finally overcomes SRNet. With regard to the progress of training in the Prop. (SRNet), only for S-UNIWARD, the progress is different from the cases of WOW and HUGO-BD. This is because there exist a few signals isolated in smooth regions in the case of S-UNIWARD. Fig. 7 shows the pixels modified in a resized stego image after each steganography is embedded at 0.4 bpp once. As shown in the top of Fig. 7(c), S-UNIWARD embeds a few stego signals into smooth regions in the original image in Fig. 7(a), and they are isolated. On the other hand, WOW and HUGO-BD does not embed stego signals into such regions as shown in Fig. 7(b) and Fig. 7(d). Since the isolated signals can be useful features for CNN to discriminate RCI from RSI_1, the training progresses auspiciously by focusing on them in normal images. However, the training is difficult in resized images. Thus, the additional embedding is conducted by the proposed preprocessing to emphasize the isolated signals. However, in the RSI_2, such signals may disappear so as to minimize the distortion after the second embedding of S-UNIWARD. In other words, after the second embedding, some of the pixel values modified by the first embedding may be returned to original pixel values. In this case, the existence of isolated signals in RSI_2 is obscure. Thus, since the obscure existence confuses whether features around the isolated signals in the smooth regions are useful or not, it is difficult for the Prop. (SRNet) which trains RSI_1 and RSI_2 to learn features. As a result, useful features are



(a) WOW (b) S-UNIWARD (c) HUGO-BD

FIGURE 6. The progress regarding ACC of the SRNet and the Prop. (SRNet) in the case where the same steganography is additionally embedded at 0.4 bpp once.



(a) Original image (b) WOW

(c) S-UNIWARD (d) HUGO-BD

FIGURE 7. The pixels modified in a resized stego image after each steganography is embedded at 0.4 bpp once. Red and blue dots denote +1 and −1 embedding, respectively.

not extracted until CNN trains to some extent, which causes the poor training until 10,000 iteration in the Prop. (SRNet). Finally, the Prop. (SRNet) can improve the detection performance after adequate training.

These results mean the proposed preprocessing can help the CNN to easily learn stego signals in the resized images. Thus, we conclude that the proposed preprocessing is useful for improving detection performance for resized stego images.

Among three steganography, we found that HUGO-BD in resized images is most undetectable steganography even in the Prop. (SRNet) although it is relatively easy to be detected in normal images. This is because the embedding strategy of HUGO-BD tries to embed stego signals into the edges that are smooth along any direction in images. In the normal images, because the edge is very clear. Therefore, because CNN should train the embedding pattern by focusing on the clear edge, stego signals are easy to be detected. However, in the resized images by the NNI, because the clear edges decrease, it is difficult for CNN to learn the embedding pattern compared with the normal images.

D. VALIDATION WITH ANOTHER STEGANOGRAPHY

We evaluate the effectiveness of the proposed preprocessing when steganography which is different from the original one is additionally embedded into the RSI_1 once. For example, it is supposed that S-UNIWARD is additionally embedded into the RCI and the RSI_1 to which WOW is originally embedded. In this case, the CNN trains the RSI_1 where only S-UNIWARD is embedded and the RSI_2 where S-UNIWARD is additionally embedded after WOW instead of RCI and RSI_1, respectively. Thus, the model learns whether WOW is embedded into the images. In this evaluation, we evaluate the detection performance for WOW, S-UNIWARD, and HUGO-BD. Additional steganography is embedded at the same payload, which is 0.4 bpp as the original steganography once. In this evaluation, the Prop. (SRNet) with WOW, S-UNIWARD, and HUGO-BD mean the proposed models that WOW, S-UNIWARD, and HUGO-BD are additionally embedded by the proposed preprocessing, respectively.

Table 9 shows the max testing ACC of the Prop. (SRNet) and SRNet for resized datasets with the combination of steganography changed. As shown in Table 9, the Prop. (SRNet) with WOW, Prop. (SRNet) with S-UNIWARD, and the Prop. (SRNet) with HUGO-BD can improve detection performance in the cases where other steganographic algorithms are originally embedded in comparison to SRNet. In particular, as we can see from the testing ACC of the Prop. (SRNet) with WOW for S-UNIWARD, the testing ACC is 74.7.%, and it is 23.1% higher in comparison to that of SRNet, which is 51.6%. This is because WOW and S-UNIWARD are based on the same basic idea that stego signals are embedded into the noisy regions to evade steganalysis whereas their detailed embedding methods are different. Thus, the proposed preprocessing is effective to detect resized stego images even when different steganography is additionally embedded.

TABLE 9. The max testing ACC of the Prop. (SRNet) and SRNet for resized datasets with the combination of steganography changed. Payload is 0.4 bpp.

Steganographic algorithm	Prop. (SRNet) with WOW	Prop. (SRNet) with S-UNIWARD	Prop. (SRNet) with HUGO-BD	SRNet
WOW	**0.856**	0.827	0.790	0.786
S-UNIWARD	0.747	**0.829**	0.731	0.516
HUGO-BD	0.635	0.654	**0.717**	0.613

Although the detection performance can be improved by additionally embedding different steganography, the ACC is lower than that in the case where the same steganography is additionally embedded. From these results, we conclude that embedding the same steganography as embedded one is promising for yielding best performance, which means the proposed preprocessing is very effective in detecting resized stego images created by known steganography. Because most existing CNN based steganalysis schemes including SRNet are based on supervised learning [7]–[9], [26], [28], they focus on detecting known steganography. However, our evaluation results show that SRNet cannot deal even with known steganography when it is embedded into resized images. Thus, additionally embedding the same steganography as the preprocessing is reasonable and useful for detecting the known steganography in resized images.

E. EFFECT ON EMBEDDING PAYLOAD

In order to inspect the effect on the embedding payload, we evaluate the detection performance of the Prop. (SRNet) and SRNet for resized images when stego signals are embedded at 0.2 and 0.4 bpp. As shown in Table 2, the difference of spatial frequencies of cover images and stego ones is not sufficiently enlarged by additionally embedding steganography once in the case where stego signals are embedded at 0.2 bpp. Hence, since we expected the detection performance of the Prop. (SRNet) is not sufficiently improved in this case, we evaluate the detection performance changing the number of additional embeddings up to 4 times. We create the Resized Stego Images after 3 embeddings (RSI_3), those after 4 embeddings (RSI_4), and those after 5 embeddings (RSI_5). Note that we additionally embed the same steganography as the one which is originally embedded in images of training datasets at the same payload in this evaluation.

Table 10 shows the max testing ACC in SRNet and the Prop. (SRNet) when the number of additional embeddings at 0.2 bpp is changed up to 4 times. As shown in Table 10, SRNet cannot train stego signals at all, and the testing ACC in SRNet is 0.500, which is random judgement. This result means that resized images created by the NNI incur terrible deterioration in detection performance of SRNet when signals are embedded at 0.2 bpp. Meanwhile, the Prop. (SRNet) can deal with such resized images by introducing the proposed preprocessing. In particular, with regard to WOW, the Prop. (SRNet) with 4 embeddings achieves testing ACC of 84.8%, which is 34.8% higher than that of SRNet. However, as we expected, the Prop. (SRNet) with 1 embedding cannot improve detection performance unlike the case of 0.4 bpp. With regard to WOW and HUGO-BD, at least 4 additional embeddings are required to adequately help the CNN train stego signals. Besides, in terms of S-UNIWARD, at least twice additionally embedding is required. Hence, additionally embedding more than 4 times is effective for detecting every steganography embedded at 0.2 bpp in our evaluation.

In order to reveal the reason of these results, we inspected the difference of average HFD_{image} between cover images and stego images in each case. Table 11 shows the difference of average HFD_{image} between 10,000 resized cover images and 10,000 resized stego ones in each case when stego signals are embedded at 0.2 and 0.4 bpp. As shown in Table 11, when the signals are at 0.2 bpp, the difference by one additional embedding is less than 0.3 in all steganography. Furthermore, the difference is enlarged as the number of embeddings increases. On the other hand, in the case where the signals are embedded at 0.4 bpp, the difference of average HFD_{image} in each steganography gets more than 0.6 by additionally embedding once. The reason why there exist these different phenomenons between two payloads is that the quantity of stego signals embedded at 0.2 bpp is small compared with that at 0.4 bpp. Thus, in the case of 0.2 bpp, the difference of HFD_{image} is hard to be enlarged, which causes the increase of additional embeddings.

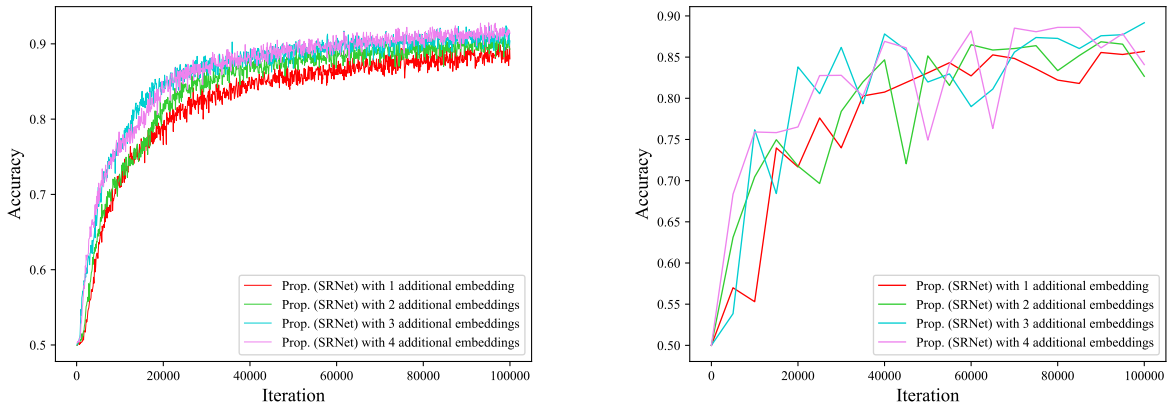
In order to further evaluate the effect of multiple embeddings in the case of 0.4 bpp, we consider that WOW is additionally embedded to RCI and RSI_1 to which WOW is originally embedded, up to 4times. Fig. 8 shows training and testing ACC in the Prop. (SRNet) when WOW is embedded at 0.4 bpp. As shown in Fig. 8(a), the progresses of training process in all cases are very similar. Furthermore, as shown in Fig. 8(b), the testing ACC scores of Prop. (SRNet) with multiple additional embeddings are also comparable to the testing ACC of Prop. (SRNet) with 1 additional embedding. Moreover, as shown in Table 11, the difference of HFD_{image} gets large as steganography is embedded. These results mean the difference of HFD_{image} between RSI_1 and RSI_2 is sufficient to detect the stego signals. Thus, the results shown in Fig. 8 mean that the number of required embedding times for improving the detection performance is at least once in the case where embedding payload is 0.4 bpp in our experiments. Although embedding more than once is also effective in the case of 0.4 bpp, the improvement seems to reach the ceiling. Therefore, we conclude that additionally embedding more than once is not entirely required in the case of 0.4 bpp. Thus, multiple steganography embeddings are required only when stego signals are embedded at a small payload. Furthermore, since the proposed preprocessing is very simple, it does not greatly increase the training time of CNN. From these results, we conclude that the proposed preprocessing can improve the detection performance in the cases of both 0.2 and 0.4 bpp without increasing computational cost on CNN.

TABLE 10. The max testing ACC in SRNet and the Prop. (SRNet) when the number of additional embeddings at 0.2 bpp is changed up to 4 times.

Model (cover images and stego ones trained in CNN)	WOW	S-UNIWARD	HUGO-BD
SRNet (RCI and RSI_1)	0.500	0.500	0.500
Prop. (SRNet) with 1 embedding (RSI_1 and RSI_2)	0.500	0.500	0.500
Prop. (SRNet) with 2 embeddings (RSI_2 and RSI_3)	0.500	0.823	0.500
Prop. (SRNet) with 3 embeddings (RSI_3 and RSI_4)	0.500	0.793	0.527
Prop. (SRNet) with 4 embeddings (RSI_4 and RSI_5)	**0.848**	**0.838**	**0.635**

TABLE 11. The difference of average HFD_{image} between 10,000 resized cover images and 10,000 resized stego ones with the number of additional embeddings changed when stego signals are embedded at 0.2 and 0.4 bpp. The bold numbers are the minimum difference of average HFD_{image} enough to improve the detection performance of CNN in our evaluation.

The number of embedding (cover images and stego ones trained in CNN)	WOW		S-UNIWARD		HUGO-BD	
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
No embedding (RCI and RSI_1)	0.090	0.285	0.084	0.252	0.128	0.345
Once (RSI_1 and RSI_2)	0.214	**0.603**	0.215	**0.622**	0.285	**0.742**
Twice (RSI_2 and RSI_3)	0.313	0.861	**0.296**	0.857	0.382	0.961
3 times (RSI_3 and RSI_4)	0.399	1.072	0.362	1.027	0.465	1.158
4 times (RSI_4 and RSI_5)	**0.465**	1.340	0.411	1.160	**0.519**	1.273



(a) Training ACC (b) Testing ACC

FIGURE 8. Training and testing ACC in the Prop. (SRNet) when WOW is additionally embedded multiple times at 0.4 bpp.

TABLE 12. The max testing ACC for images resized by the NNI in the case where SRNet, Xu-Net, and SRM are used.

Model	WOW		S-UNIWARD		HUGO-BD	
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
SRNet	0.500	0.786	0.500	0.516	0.500	0.613
Prop. (SRNet)	**0.848**	**0.856**	**0.838**	**0.829**	**0.635**	**0.717**
Xu-Net	0.505	0.514	0.502	0.510	0.504	0.524
Prop. (Xu-Net)	**0.549**	**0.621**	**0.534**	**0.576**	**0.545**	**0.552**
SRM	0.565	0.623	0.544	0.609	0.538	0.595
Prop. (SRM)	**0.652**	**0.712**	**0.649**	**0.711**	**0.589**	**0.666**

F. COMPATIBILITY WITH OTHER CONVENTIONAL STEGANALYSIS

In order to confirm effectiveness of the proposed preprocessing for other steganalyzer, we also evaluate Xu-Net and SRM as a CNN-based steganalyzer and a feature extraction based steganalyzer, respectively. Table 12 shows the max testing ACC for images resized by NNI in the case where SRNet, Xu-Net, and SRM are used. Prop. (Xu-Net) and Prop. (SRM) mean that the methods in which the proposed preprocessing is combined with Xu-Net and SRM, respectively. Note that additionally embedding is conducted 4 times for the case of 0.2 bpp although it is performed once for the case of 0.4 bpp. This is because the detection performance for WOW at 0.2 bpp is not improved until stego signals are additionally embedded 4 times. As shown in Table 12, the proposed preprocessing is effective in improving the detection performance of all of three conventional schemes.

In terms of Xu-Net, according to the literature [23], Xu-Net achieves about 80% ACC for normal images in which S-UNIWARD is embedded at 0.4 bpp. However, the ACC of Xu-Net for S-UNIWARD in resized images is 51%. This result demonstrates that the effect of the NNI also degrades the detection performance of a basic CNN-based steganalysis. When the proposed preprocessing is incorporated with Xu-Net, the testing ACC of Prop. (Xu-Net) for S-UNIWARD at 0.4 bpp is 6.6% higher than that of Xu-Net. Furthermore, in the case of S-UNIWARD at 0.2 bpp, the testing ACC of Prop. (Xu-Net) is 3.2% higher than that of Xu-Net. Similarly, Prop. (Xu-Net) can also improve the performance for WOW and HUGO-BD. However, in comparison to Prop. (SRNet),

Prop. (Xu-Net) is not very effective in detection of the images resized by NNI in the cases of both 0.2 and 0.4 bpp. In particular, the ACC tends to be lower than that of the other schemes in the case of 0.2 bpp. This is because the stego signals emphasized by the proposed preprocessing are not learned effectively during the training of Xu-Net. Furthermore, it is possible that the useful stego signals are suppressed because of average pooling layer used after each convolutional layer in Xu-Net. Meanwhile, in SRNet, the average pooling is not introduced in the front layers to avoid such disadvantage. In addition to that, residual learning is not used in Xu-Net unlike SRNet. In residual learning, the architecture called "residual shortcut" is involved. Introducing residual shortcut helps learn the parameters in upper layers of deep networks although the parameters in upper layers are typically hard to be learned. In particular, the training of stego signals in resized images is difficult compared with normal images. Thus, in resized images, our results show that even features emphasized by the proposed preprocessing cannot efficiently be learned without residual learning. From these results, we conclude that the proposed preprocessing is not very compatible with Xu-Net although the performance is improved.

As for SRM, the proposed preprocessing is effective in all cases. Particularly, for S-UNIWARD at 0.2 bpp, the testing ACC of Prop. (SRM) is 10.5% higher than that of SRM. Thus, the results demonstrate that the proposed preprocessing is applicable to not only CNN-based steganalysis but also a traditional feature extraction based steganalysis.

G. VALIDATION OF IMAGE DOWNSAMPLING METHOD

Although we utilize the NNI to show the shortcoming of SRNet, recent image downsampling methods are sophisticated and resize images so that the looks of resized images are smooth. In other words, high frequency does not occur much in the resized images by a sophisticated image downsampling method. Thus, we evaluate the effect by the kind of image downsampling algorithms, namely the NNI and the sophisticated one. We use Bilinear [29] and Bicubic interpolation [30] as the sophisticated image downsampling method. In Bilinear interpolation, each pixel of a image is complemented on the basis of 4 pixels around the complemented one. On the other hand, in Bicubic interpolation, each pixel of a image is complemented on the basis of 16 pixels around the complemented one. These interpolation are known as the methods which can minimize loss of information contained in a image. In fact, some researchers adopt Bicubic interpolation to resize images when they create reasonable image dataset [31], [32].

In this evaluation, we only use WOW at 0.4 bpp. Table 13 shows the max testing ACC of SRNet [9] for cropped images and resized images created by each downsampling method. As we can see from Table 13, the only NNI degrades the detection performance of SRNet, which is state-of-the-art CNN-based steganalysis. In comparison to cropping, the testing ACC in most cases of Bilinear and Bicubic are improved rather than be degraded except for the case of Bicubic for HUGO-BD. This is because the Bilinear and the Bicubic interpolations resize images to make sure that the changes of pixels among neighboring pixels are smooth, which means the resized images with low frequencies are created. Table 14 shows the average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images made by each image downsampling method. As shown in Table 14, the $\mathrm{HFD}_{\mathrm{image}}$ of RCI in Bilinear and Bicubic are low compared with that of Nearest, which means that the spatial frequencies of RCI created by sophisticated methods are low. Furthermore, the differences of $\mathrm{HFD}_{\mathrm{image}}$ between RCI and RSI_1 in the cases of Bilinear and Bicubic are large compared with the case of Nearest. These results mean that the differences of spatial frequencies between resized cover images and resized stego ones are noticeable in the case of sophisticated methods as with cropped images. Because statistical modeling on such images is relatively easy, CNN can find the presence of stego signals because of the obvious difference. From these results, we conclude that the Bilinear and the Bicubic interpolations are rarely used by attackers because they want to make the detection of stego signals difficult. Since we can easily assume that attackers intentionally use the NNI to evade the detection, countermeasures which are intended for the NNI are required. However, there has still been no studies which investigate useful countermeasures. This fact motivates us to devise the proposed preprocessing for the NNI. Thus, from the perspective of security, we regard this work as a method which is intended for images resized by the NNI.

TABLE 13. The max testing ACC of SRNet for cropped images and resized images created by each downsampling method. (Payload is 0.4 bpp in all steganographic algorithms).

Downsampling method	WOW	S-UNIWARD	HUGO-BD
Cropping	0.853	0.847	0.874
Nearest	**0.786**	**0.516**	**0.613**
Bilinear	0.965	0.938	0.932
Bicubic	0.896	0.886	0.854

TABLE 14. The average of $\mathrm{HFD}_{\mathrm{image}}$ of the 10,000 resized images made by each image downsampling method.

Downsampling method	RCI	RSI_1	Difference
Nearest	99.501	99.786	0.285
Bilinear	22.772	23.535	0.763
Bicubic	39.790	40.358	0.568

H. EFFECTIVENESS FOR OTHER DOWNSAMPLING METHODS

As described in Section V-G, the other downsampling methods except for the NNI would rarely be used by attackers. However, since it is possible that images resized by other downsampling methods are used as cover ones, the generalization of the proposed preprocessing is important for future work of steganalysis. Therefore, we evaluate whether the proposed preprocessing is applicable to other downsampling methods. Table 15 shows that the max testing ACC of SRNet

TABLE 15. The max testing ACC of SRNet and the Prop. (SRNet) for images resized by each downsampling method. (Payload is 0.4 bpp in all steganographic algorithms).

Method	Model	WOW	S-UNIWARD	HUGO-BD
Nearest	SRNet	0.786	0.516	0.613
	Prop. (SRNet)	**0.856**	**0.829**	**0.717**
Bilinear	SRNet	0.965	0.938	0.932
	Prop. (SRNet)	**0.966**	**0.979**	**0.959**
Bicubic	SRNet	0.896	0.886	0.854
	Prop. (SRNet)	**0.940**	**0.950**	**0.900**

and the Prop. (SRNet) for images resized by each downsampling method. Payload is 0.4 bpp in all steganographic algorithms. As shown in Table 15, the detection performance is improved in all cases. Thus, the results demonstrate that the proposed preprocessing is also applicable to images resized by other downsampling methods besides the NNI.

VI. LIMITATION AND FUTURE WORK

In this work, the key used for additional embedding must be the same as the key used for the first embedding to improve the detection performance. In other words, this situation means that attackers create stego images with the same key. The ideal scenario is the case where the first embedding is conducted with different keys because it should be regarded as operations by attackers. However, to the best of our knowledge, there is no useful study and effective solution to resized stego images even in the simple situation. In order to devise the solutions, we must conduct academic investigation in the simple situation in the first place. Furthermore, since there exist studies [33] and [34] which suppose the scenario where stego signals are always embedded with the same key, such scenario is not unlikely. Therefore, in this work, we conduct experiments in the simple situation where the same key is always used for different images and additionally embedding. We reveal that there exist the cases where the detection performance for resized stego images can be improved, which is important for future work of steganalysis and information forensics. At this stage, using the same key is required to improve detection performance. Thus, we think that our work is the first step of study for improving detection performance for resized stego images. In order to realize more practical methods, the keys used by attackers or locations of modified pixels (payload locations) must be estimated. In literature [34], an estimation of payload locations has been studied, which is promising work to provide clues that lead to the recovery of the embedding key. By applying that estimation scheme to our work, more practical schemes which yield the equivalent to the effect on our results may be devised in the future. As future work, we plan to research such feasibility and countermeasures in practical situations on the basis of results of this work.

VII. CONCLUSION

In this article, we have proposed a preprocessing by using multiple steganography for intentional image downsampling on CNN-based steganalysis. Whenever resized images are trained in CNN or inspected whether they are stego ones or not, steganography is applied to them by the proposed preprocessing. Since the proposed preprocessing is very simple, it does not greatly increase the training time of CNN. Our evaluation results demonstrate the proposed preprocessing is useful to deal with resized stego images created by the NNI although the state-of-the-art CNN based steganalysis cannot deal with them at all. Furthermore, the proposed preprocessing is also effective even when different steganography is additionally embedded. Although the proposed preprocessing is useful, we should devise a more practical way which yields the comparable effectiveness to it. This work would be a first step of countermeasures against resized stego images and contributes to future studies in steganalysis and information forensics.

REFERENCES

[1] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Process.*, vol. 90, no. 3, pp. 727–752, Mar. 2010, doi: 10.1016/j.sigpro.2009.08.010.
[2] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," *Image Commun. Signal Process.*, vol. 65, pp. 46–66, Jul. 2018.
[3] P. Bak, J. Bieniasz, M. Krzeminski, and K. Szczypiorski, "Application of perfectly undetectable network steganography method for malware hidden communication," in *Proc. 4th Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2018, pp. 34–38.
[4] K. Choudhary, "Image steganography and global terrorism," *Int. J. Sci. Eng. Res.*, vol. 3, pp. 1–12, Jul. 2012.
[5] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.
[6] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
[7] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Proc. Signal Inf. Process. Assoc. Summit Conf.*, pp. 1–4, 2014.
[8] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," *Media Watermarking, Secur., Forensics*, vol. 9409, p. 94090J, Mar. 2015.
[9] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019.
[10] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Dec. 2012, pp. 234–239.
[11] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, pp. 1–13, Dec. 2014.
[12] D. Lerch-Hostalot and D. Megías, "Detection of classifier inconsistencies in image steganalysis," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jul. 2019, pp. 222–229.
[13] J. De Jesus Serrano Perez, M. S. Rosales, and N. Cruz-Cortés, "Universal steganography detector based on an artificial immune system for JPEG images," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1896–1903.
[14] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
[15] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Berlin, Germany: Springer, 2000, pp. 1–15.
[16] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. Int. Workshop Inf. Hiding*, in Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 6958. Berlin, Germany: Springer, 2011, pp. 59–70.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[19] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 705–720, Dec. 2010.

[20] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10437–10453, 2018.

[21] J. Kodovsky and J. Fridrich, "Effect of image downsampling on steganographic security," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 752–762, May 2014.

[22] Y.-H. Tang, L.-H. Jiang, H.-Q. He, and W.-Y. Dong, "A review on deep learning based image steganalysis," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 1764–1770.

[23] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.

[24] *Binghamton Website*. Accessed: Nov. 2019. [Online]. Available: http://dde.binghamton.edu/download

[25] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimedia Tools Appl.*, vol. 77, no. 9, pp. 10437–10453, 2018.

[26] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Efficient feature learning and multi-size image steganalysis based on CNN," no. 2, pp. 1–10, 2018, *arXiv:1807.11428*. [Online]. Available: http://arxiv.org/abs/1807.11428

[27] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[28] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

[29] P. R. Smith, "Bilinear interpolation of digital images," *Ultramicroscopy*, vol. 6, no. 1, pp. 201–204, Jan. 1981.

[30] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.

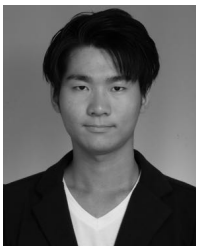
[31] S. Tan, H. Zhang, B. Li, and J. Huang, "Pixel-decimation-assisted steganalysis of synchronize-embedding-changes steganography," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 7, pp. 1658–1670, Jul. 2017.

[32] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Efficient feature learning and multi-size image steganalysis based on CNN," 2018, *arXiv:1807.11428*. [Online]. Available: http://arxiv.org/abs/1807.11428

[33] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch," *Electron. Imag.*, vol. 2016, no. 8, pp. 1–11, Feb. 2016.

[34] T.-T. Quach, "Cover estimation and payload location using Markov random fields," *Proc. SPIE*, vol. 9028, Feb. 2014, Art. no. 90280H.

HIROYA KATO (Graduate Student Member, IEEE) was born in Gunma, Japan, in 1994. He received the B.E. and M.E. degrees from Keio University, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree. His research interest includes security and privacy for the Internet of Things (IoT). He is a member of IEICE.



KYOHEI OSUGE was born in Kanagawa, Japan, in 1995. He received the B.E. and M.E. degrees from Keio University, in 2018 and 2020, respectively. His research interest includes security and privacy for the Internet of Things (IoT). He is a member of IEICE.

SHUICHIRO HARUTA (Member, IEEE) was born in Saitama, Japan, in 1992. He received the B.E., M.E., and Ph.D. (engineering) degrees from the Department of Information and Computer Science, Keio University, Yokohama, Japan, in 2015, 2017, and 2020, respectively. He was a Research Associate with Keio University, from 2017 to 2018. His research interest includes security and privacy for the Internet of Things. He is a member of IEICE.

IWAO SASASE (Senior Member, IEEE) was born in Osaka, Japan, in 1956. He received the B.E., M.E., and D.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1979, 1981, and 1984, respectively. From 1984 to 1986, he was a Postdoctoral Fellow and a Lecturer of Electrical Engineering with the University of Ottawa, Ottawa, ON, Canada. He is currently a Professor of Information and Computer Science with Keio University. He has authored more than 293 journal articles and 441 international conference papers. His research interests include modulation and coding, broadband mobile and wireless communications, optical communications, communication networks, and information theory. He granted 45 Ph.D. degrees to his students in the above field. He is a Fellow of the Institute of Electronics, Information, and Communication Engineers (IEICE) and a member of the Information Processing Society of Japan. He received the 1984 IEEE Communications Society (ComSoc) Student Paper Award (Region 10), the 1986 Inoue Memorial Young Engineer Award, the 1988 Hiroshi Ando Memorial Young Engineer Award, the 1988 Shinohara Memorial Young Engineer Award, the 1996 IEICE of Japan Switching System Technical Group Best Paper Award, and the WPMC2008 Best Paper Award. He is also serving as the Vice President of IEICE. He served as the President of the IEICE Communications Society, from 2012 to 2014. He was the Board of Governors Member-at-Large, from 2010 to 2012; the Japan Chapter Chair, from 2011 to 2012; the Director of the Asia Pacific Region, from 2004 to 2005, and the Society of Information Theory and Its Applications, Japan, from 2001 to 2002; the Chair of the Satellite and Space Communications Technical Committee, from 2000 to 2002, of IEEE ComSoc, the Network System Technical Committee, from 2004 to 2006, and the Communication System Technical Committee, from 2002 to 2004, of the IEICE Communications Society; and the Vice President of the Communications Society, from 2004 to 2006.