Signal Processing: Image Communication 113 (2023) 116927

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

ACGIS: Adversarial Cover Generator for Image Steganography with Noise Residuals Features-Preserving

Junxue Yang, Xin Liao *

College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

ARTICLE INFO

Keywords:
Steganography
Adversarial cover
Siamese generative network
Sub-regions noise residuals features

ABSTRACT

Recent works with the technique of adversarial example have been bringing the possibility of effectively resisting the machine learning-based steganalyzers. Nevertheless, these methods likely introduce unexpected artifacts and destroy the statistics when adding adversarial perturbations. In this paper, under the assumption of similarity between the noise residuals of normal image sub-regions, we propose a Siamese generator to learn and preserve sub-regions noise residuals features for minimizing the impact of adversarial perturbations on similarity. The cover and stego sub-regions pairs are used as the input, which incorporates steganography domain knowledge to further encourage the generator to yield the more favorable adversarial covers. Moreover, during interactive training with steganalyzer, using a random embedding strategy to replace the specific steganographic algorithm saves training time and improves the generalization. We can employ the trained generator to produce numerous adversarial covers, cooperating with the existing steganographic methods to embed secret messages achieving much safer steganography. Security analysis and experiments show that the generated adversarial covers are superior in terms of quality and security.

1. Introduction

Image steganography is the science and art of sheltering confidential information within images. Specifically, it takes advantage of the inherent redundancy of images to hide secret information in images without destroying the image quality as much as possible, to achieve covert communication. An image that seems natural to the casual observer probably conceals secret messages only extracted by the expected receiver, utilizing the psychology that seeing is believing to deceive the attacker. It is generally implemented by imperceptibly varying the pixel values (in spatial domain) or DCT coefficients (in JPEG domain) [1]. Steganalysis, the countermeasure of steganography, is the art of identifying whether a secret message is hidden in the medium, i.e., determining whether the medium belongs to the cover class or the stego class, although in some cases it may also include extraction and/or destruction of the data [2]. Steganalysis involves two major types of techniques: visual analysis and statistical analysis. Visual analysis tries to reveal the presence of covert communication through inspection, either with the naked eye or with the assistance of a computer. Statistical analysis is more powerful and mainstream since it reveals tiny alterations in an image's statistical behavior caused by steganographic embedding [3].

As opponents to each other, steganography and steganalysis have been learning and promoting from the mutual competition. The earliest steganographic schemes are implemented by improving the substitute mode of non-significant bits. All of them indiscriminately embed messages into pixels ignoring image features, which alters the statistical properties [4]. Accordingly, the manipulated traces yielded by these schemes can be effectively detected by the specific steganalyzers based on low-order statistics [5]. Over time the growing steganography, the steganographic methods are proposed under the framework of distortion minimization, namely content-adaptive steganography. In this framework, the main task of steganography is to design a reasonable heuristically-defined distortion function, and then utilizes STCs (syndrome-trellis codes) [6] to achieve near the minimal distortion embedding with a given payload. Examples include WOW (wavelet obtained weights) [7], S-UNIWARD (spatial-universal wavelet relative distortion) [8], HILL (high-pass, low-pass, lowpass) [9], and CMDs (clustering modification directions) [10]. They tend to hide secret data into the regions with rich texture preserving image statistics as much as possible [11]. As the countermeasure, more advanced and higher-dimensional steganalysis features, such as SRM (spatial rich model) [12], are designed to capture embedding artifacts, and the ensemble classifier [13] is employed for effective training. Recently, steganalyzers based on deep learning [14–16] have been attracting increasing attention. The performance of these steganalyzers has exceeded the steganalyzers based on hand-crafted features, and the precision is even close to 90% at the representative payload of 0.4 bpp (bits per pixel), which brings a great challenge to steganography. Although

* Corresponding author.
E-mail address: xinliao@hnu.edu.cn (X. Liao).

https://doi.org/10.1016/j.image.2023.116927
Received 21 April 2022; Received in revised form 28 October 2022; Accepted 18 January 2023
Available online 26 January 2023

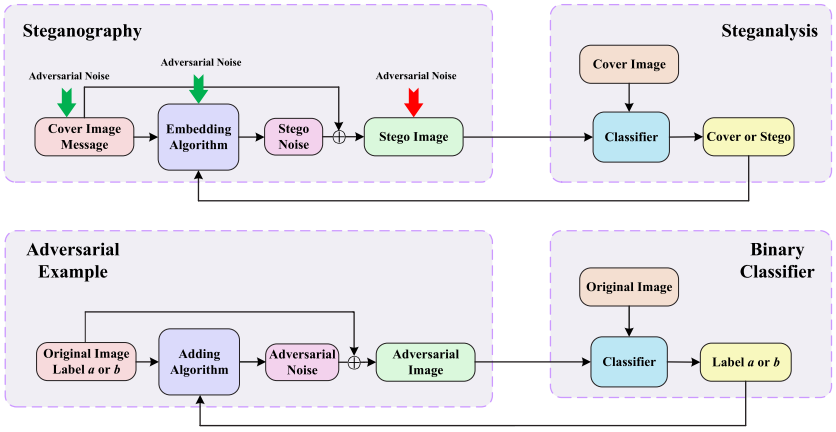

Fig. 1. The process of steganography and adversarial example. The swallow-tailed arrows indicate the position that adds adversarial noise, and their color, green or red, means where adding perturbations is workable or unworkable.

the subsequent steganographic methods [17–26] based on GAN (generative adversarial networks) are proposed, their performance against CNN-based steganalyzers has not been obviously improved. Hence, it is necessary to improve the steganographic schemes to enhance performance.

In fact, steganalysis is usually viewed as a binary classification problem, classifying the images before and after embedding, i.e., cover images and stego images, which can be solved by employing machine learning and neural networks. Yet machine learning systems, including neural networks, have been shown to be sensitive to adversarial example [27,28]. Adversarial example is a technique that adds infinitesimal perturbations to the input to make the classifier output an incorrect label. Ref. [27] puts forward that adversarial example can be thought of as "accidental steganography". Analogously, the embedding process is usually regarded as adding extremely low amplitude steganographic signals to the image. Fig. 1 describes the process of steganography and adversarial example and three possible locations indicated by the swallow-tailed arrows to add perturbations to steganography. Steganography can hide messages into the cover image by designing an embedding algorithm to obtain the corresponding stego image that neither the human eyes nor steganalyzers cannot distinguish [29]. The adversarial image is devised via adding perturbations in the original image with the ability of imperceptibility and misclassification. Comparisons of the process and performance requirements (imperceptibility and misclassification) of steganography and adversarial example show that there are plenty of correspondences between them, which provides fresh research ideas.

Considering the steganographic process, an intuitional way to combine steganography with adversarial example is to add adversarial perturbations to the stego image. However, such behavior will cause the receiver to extract incorrect secret messages. Two additional feasible ways include that utilizes adversarial example to construct an adversarial cover image and adjust the distortion cost. Generally speaking, the former has a more obvious misclassification ability than the latter and therefore safer. In the way of constructing an adversarial cover, Zhang et al. [30] construct the robust adversarial one via iteratively adding adversarial perturbations to cover image based on the gradients. Different from Zhang's method, Zhou et al. [31] design the adversarial one via adding adversarial perturbations to cover image according to adversarial learning. The trained generator can quickly produce adversarial covers suitable for steganography, which has great advantages in time consumption. Tang et al. [32] develop the second way of fine-tuning the costs of image elements modifications according to the gradients under the framework of distortion minimization. On this basis, Bernard et al. [33] utilize adversarial embedding and game theory to design an adaptive cost function.

In this paper, we follow the study of constructing an adversarial cover. The existing works about adversarial cover usually add adversarial perturbations ignoring the relationships between the noise residuals of image sub-regions, which likely introduces unexpected artifacts and destroys the statistical properties. With these inadequacies come increased risk in adversarial cover image quality and security. Based on the above considerations, an end-to-end model is proposed to quickly produce the visually-pleasing and enough-secure adversarial cover. Our model contains two modules: a generator and a steganalyzer. The generator adopts the Siamese, a CNN (convolutional neural network)-based architecture that can mine the similarity between the input pair, which is used to capture and preserve image sub-regions noise residuals relationships when generating adversarial cover, guaranteeing visual quality and security. Furthermore, the proposed model combines with the steganalysis network to improve security further by absorbing the game theory. The main contributions are summarized as follows:

(1) We propose a novel method of constructing adversarial covers, in which the Siamese generative network can capture and preserve the image sub-regions noise residuals relationships to achieve high quality and security construction.

(2) Aided by the similarities between steganography and adversarial example, the stego images acquired by the content-adaptive steganographic algorithm participate in training to further promote the Siamese generator to learn features that can keep the relationships better.

(3) Numerous experiments demonstrate that the proposed method is compatible with different steganographic algorithms and payloads and can still generate adversarial covers with high quality and security in contrast with existing works without retraining.

The remainder of this paper is organized as follows. In Section 2, we summarize the GAN-based steganographic works. And the usual preliminaries of the proposed model are introduced, including noise residuals computation and adversarial example. The basic idea of our model is described in Section 3, and a novel adversarial cover generation framework is designed. Section 4 presents investigative experiments and analysis aimed at verifying the performances. Finally, the conclusions are made in Section 5.

2. Preliminaries

In this section, we first detail the GAN-based steganographic works. Also, we present the usual preliminaries of the proposed model from two aspects: noise residuals computation and adversarial example.

2.1. GAN-based image steganography

During adversarial training, GAN-based steganographic methods enhance the ability to resist steganalysis from the perspective of generating cover images [17–19], embedding costs [20,21], and stego images [22–26].

The methods of generating cover images: Volkhonskiy et al. [17] first propose a steganographic model SGAN based on DCGAN [34]. SGAN contains a generator and two discriminators. The purpose of adversarial learning between the generator and two discriminators is to make the generated cover images closer to natural images and more suitable for steganography, respectively. However, the training process of SGAN is not stable, and the generated cover images are visually poor. On the foundation of SGAN, SSGAN [18] adopting WGAN [35] and VAE-SGAN [19] adopting VAE-GAN [36] have also been presented successively. Although compared to SGAN, SSGAN and VAE-SGAN can obtain a faster converge speed and visually better cover images, the generated cover image based on random noise is still unnatural, and the security is not improved much.

The methods of generating embedding costs: ASDL-GAN is initially introduced by Tang et al. [20] to imitate the process of content-adaptive steganography, which consists of a generator, a ternary embedding simulator, and a discriminator. In ASDL-GAN, the generator takes the cover image as input and automatically learns to generate embedding costs. Then the ternary embedding simulator can obtain the simulated stego image according to the generated embedding costs. The discriminator is used to distinguish the cover image from the stego image to improve security. Nevertheless, the non-differentiability of the simulator leads to slow convergence of ASDL-GAN, and its security has not exceeded the traditional content-adaptive steganographic methods. Yang et al. [21] make partial improvements under the former framework, which use the double-tanh function instead of the previous simulator to speed up model training and design a finer generator based on U-Net to enhance security. But it is only slightly more secure than traditional methods.

The methods of generating stego images: Utilizing generative adversarial networks to embed and extract secret information is first proposed by Hayes et al. This steganographic model [22] defines a three-way game of encoder, decoder, and steganalyzer, which are used for information hiding, information extraction and steganalysis, respectively. However, there are visual differences between stego images generated by the model and cover images that are visible to the naked eye. Zhu et al. [23] construct a noise layer between the encoder and decoder to model the noise for improving robustness, considering that stego images may be damaged during communication. Nevertheless, the maximum hiding capacity of Ref. [23] can only reach about 0.2 bpp, which is caused by the large dimension when processing secret information into a tensor. Moreover, the proposed model still has weaknesses in the anti-detection performance. Zhang et al. [24] improve the hiding capacity by organizing binary information bits into a 3-D tensor concatenated with the image tensor in the encoder. Two recent works, spatial attention-based ABDH [25] and channel attention-based CHAT-GAN [26], improve security and extraction rate using advanced network modules. However, due to the information loss of the deep model itself, the methods of generating stego images cannot guarantee the exact extraction of secret information, and their security is not significantly improved compared with the traditional content-adaptive steganographic methods.

2.2. Noise residuals computation

As mentioned earlier, both adversarial example and steganography can be essentially treated as adding weak signals to the image. Instead of the original pixels, it is wiser to model the noise residuals, which preferably captures the faint perturbations in the image. Steganalysis has been giving full scope to such an idea [4,12]. For an image $X = (x_{ij})^{h\times w}$, h and w are the height and width of the image, and the noise residuals $R = (r_{ij})^{h\times w}$ are usually calculated from a pixel predictor, i.e.,

$$r_{ij} = Pre(\mathcal{N}_{ij}) - c \times x_{ij} \tag{1}$$

Where $\mathcal{N}_{ij}$ represents a local neighborhood of the pixel x_{ij}, c is the residual order, and $Pre(\cdot)$ is a predictor of $c \times x_{ij}$ defined on $\mathcal{N}_{ij}$. In effect, the predictor is performed by the filters. Also, the noise residuals can be reformed as

$$R = X \otimes F \tag{2}$$

Where $\otimes$ is the convolution operator, F denotes the filter.

There is a wide selection of filters (linear or nonlinear). We can employ them to generate different noise residuals, which capture rich and representative statistic features. SRM is a classical steganalysis feature set, and its success cannot be divorced from the diversity of noise residuals. The SRM has the 30 basic liner filters corresponding to 7 residual classes. All filters are with the maximum kernel size of 5×5, padding zeros if the size is insufficient. In our implementation, we utilize SRM to model the noise residuals and further optimize the noise residuals by exploring the possibility of the Siamese network under the assumption of similarity between the noise residuals of normal image sub-regions.

2.3. Adversarial example

Adversarial example can easily cheat machine learning systems. For example, an image has once been classified as label a by the model, yet classified as label b after adding subtle changes undetected by the human eyes. It is the high-dimensional input and linear nature of the model, rather than over-fitting, that cause the existence of adversarial example [27]. By extension, in high-dimensional space, subtly altering the input is sufficient to cause significant changes through matrix-chain multiplication. The condition, high-dimensional input, of producing the adversarial example can be satisfied due to the image having a sufficient dimension. Generally, given a targeted classifier $\Psi(\cdot)$ and input image X^o, the basic construction procedure of adversarial example X^a can be formulated as

$$\Psi(X^a) \neq t, \quad s.t. \| X^a - X^o \|_p \leq \varepsilon \tag{3}$$

Where t is the ground truth of X^o, ε controls the range of difference to prevent the added perturbation from being too obvious. And $\| \cdot \|_p$ denotes an appropriately chosen norm function, such as ℓ_0, ℓ_2 and ℓ_∞. Then obtaining the solution of X^a by

$$\begin{aligned} &X^a = \arg\max f(X', t) \\ &X' : \| X' - X^o \|_p \leq \varepsilon \end{aligned} \tag{4}$$

In Eq. (4), $f(\cdot)$ represents the loss function of classifier $\Psi(\cdot)$, and X' is one of the solutions that satisfy the difference is less than or equal to ε.

3. The proposed ACGIS model

In this section, we propose the adversarial cover generator for image steganography with noise residuals features-preserving (ACGIS). Subsequently, we introduce the model details, including the network structures and loss functions.

3.1. Basic idea

Considering that noise residuals are more sensitive to weak disturbances than the original pixel values, we assume that there is an intrinsic similarity between the sub-regions noise residuals of the normal image. Nevertheless, adversarial example adds perturbations to the normal image, which has an influence on intrinsic similarity and makes obvious differences. To construct favorable adversarial covers,

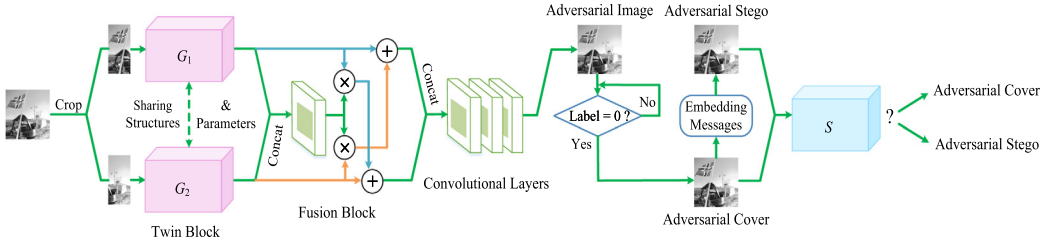

Fig. 2. The schematic illustration of the proposed ACGIS. The ACGIS consists of a generator containing the twin block with shared structures and parameters, the fusion block and convolution layers, and a steganalyzer. The example image is "55.pgm" from BOSSBase. In the training phase, the generator is fed with original cover and stego sub-regions pairs, and the input of the steganalyzer is the generated adversarial cover and its stego obtained by a random embedding strategy. In the testing phase, only the original cover is fed into the trained generator to generate the adversarial cover.

we should minimize the impact of adversarial perturbations on similarity. Correspondingly, the Siamese network [37] is typically prescribed to mine the similarity between the input pair, equipped with the contrastive loss to further optimize the extracted features. In view of these, the twin structure is introduced into the generator to preferably mine the image sub-regions noise residuals relationships for constructing adversarial ones close to normal images. Due to the structural property and task of Siamese, pairwise inputs and labels are required during training. And because the content-adaptive steganography tends to embed secret data into the regions with rich texture preserving image statistics as much as possible, in our implementation, let the left and right halves of the cover image be similar (label 0, a "genuine pair") and that of the stego image, acquired by content-adaptive steganographic algorithm, be dissimilar (label 1, an "impostor pair"). Involving stego is to incorporate steganography domain knowledge, which encourages the generator to produce the adversarial images preserving sub-regions noise residuals relationships as well as possible. Besides, we combine with game theory, utilizing a CNN-based steganalyzer to further improve security.

3.2. Network design

The schematic illustration and network architecture details of the proposed ACGIS model are shown in Fig. 2 and Table 1, respectively. The ACGIS consists of two modules: a steganalysis network Ste and a generative network that includes the twin block G_1 and G_2 with shared structures and parameters, the fusion block Fus, and convolutional layers $Covs$. The generative network is trained on cover and stego sub-regions pairs along with a binary pairwise label marking whether the two sub-regions belong to cover or stego. It aims at boosting the authenticity and undetectability of generated adversarial images. The steganalysis network is dedicated to distinguishing the adversarial cover before and after embedding. Considering that there are plenty of steganographic algorithms available, it seems infeasible to train just one or retrain each one from scratch during interactive training with steganalyzer. Thus, the stego S_{C^a} corresponding to the adversarial cover C^a is obtained by randomly embedding the value (0, −1, +1) in proportion. It not only reduces the time consumption but also improves the generalization. For example, when the payload is 0.4 bpp, the proportion is 0.8 : 0.1 : 0.1 [4]. For a better reading, the concrete training steps are listed in Algorithm 1. Once the model finishes training, we can apply the generator to produce plenty of adversarial covers. Afterwards, cooperating with the existing steganographic methods to achieve the embedding and extraction of secret information. Combined with these steganographic algorithms also can guarantee the correct extraction of confidential messages.

Generator: The generator contains the twin block, the fusion block, and convolutional layers. The twin block shares structures and parameters, each comprising the SRM, a residual block with CBAM (convolutional bottleneck attention module) [38], the convolution–deconvolution group based on U-Net [39]. In the first layer, we make the best of the learnable SRM to model the noise residuals for better seizing the relationships between image left and right sub-regions noise residuals. The weights are initialized with the 30 basic filters. The second layer is a residual block with CBAM that integrates channel and spatial attention mechanisms successively. Such design accelerates model convergence while increasing expressiveness. Similar to the U-Net, the convolution and deconvolution group incorporates seven convolutional layers and the same number of deconvolutional layers. The feature maps yielded by the former convolution layers are reused in the latter deconvolution layers to mix multi-scale features. Then, these features extracted by two sub-nets are further fused through the fusion block, encouraging rich interactions between high-level noise residuals features in image sub-regions, which can increase the intensity of adversarial perturbations as much as possible under the constraint of the generative loss. Finally, followed by three convolutional layers, the adversarial covers, the same size as the input, are obtained.

Algorithm 1: Training steps of ACGIS

Input: Original image X^o (including cover C^o and stego S^o),
generator G,
steganalyzer Ste
Output: Trained generator G_{tra}
Parameter: Learning rate lr, generative loss L_G, steganalysis loss L_S
Training Variables: Θ_G, Θ_S

```
Initialize generator with random weights.
Initialize steganalyzer with pre-trained weights for 200 epochs on cover-stego pairs.
while step < max_steps do
  for i = 1 to num_iter do
    Crop X^o into the left and right halves X_l^o, X_r^o along with a binary pairwise label 0 or 1.
    Generate adversarial image X^a = Covs(Fus(G_1(X_l^o), G_2(X_r^o))).
    loop:
    if label == 0 then
      Acquire the corresponding adversarial stego S_{C^a} = Emd(C^a), C^a ∈ X^a.
    else
      Goto loop.
    end if
    Feed C^a and S_{C^a} into the steganalyzer Ste to classify.
    Update Θ_S = Θ_S − lr × ∂L_S/∂Θ_S.
    Update Θ_G = Θ_G − lr × ∂L_G/∂Θ_G.
  end for
end while
return Trained generator G_tra
```

Steganalyzer: The steganalysis network distinguishes the adversarial cover before and after randomly embedding. Here, we select YeNet, a characteristic CNN-based steganalyzer [15], as the discriminator to promote the adversarial cover to possess a more powerful misclassification ability. YeNet can well simulate and optimize the paradigm of traditional steganalyzers by utilizing domain knowledge and CNN, which consists of nine convolutional layers, a fully connected layer,

Table 1
The network architecture details of the proposed ACGIS. SRMConv indicates the weights of the convolutional layer are initialized with the 30 SRM filters. Conv and Deconv mean the common convolution and deconvolution. FC is the full connection, BN is the batch normalization, and MP is the mean pooling. Also, Conc represents the concatenation, Mul stand for the multiplication, and Add means the addition.

Module name	Block/Layer name	Operation information ($operation1-...$)	Convolution/Deconvolution kernel ($width\times height\times(depth1,...)/(stride1,...)$)
Generator	Twin Block	SRMConv-BN-ReLU -Residual Block with CBAM -Seven Times Conv-Seven Times Deconv	$3\times3\times(30, 64, 128)/(1, 2)$ $5\times5\times(30, 64, 128)/(1, 2)$
	Fusion Block	Conc-Conv-Two Times (Mul-Add)-Conc	$3\times3\times128/2$
	Convolutional Layers	Two Times (Conv-BN-ReLU)-Conv	$3\times3\times(1, 64, 128)/1$
Steganalyzer	Layer I	SRMConv-TLU-BN	$5\times5\times30/1$
	Layer II	Four Times (Conv-ReLU)	$3\times3\times(16, 30, 32)/(1, 3)$
	Layer III	Four Times (Conv-ReLU-MP)	$2\times2\times30/2$ $3\times3\times(30, 32)/(1, 2)$ $5\times5\times32/1$
	Layer IV	FC-Softmax	/

and the Softmax. The first convolutional layer is initialized with the 30 SRM filters. Moreover, to adapt to the distribution of the embedding signals, a new activation function named TLU (truncated linear unit) is designed and introduced after the first convolutional layer, and for the others, the ReLU is employed.

3.3. Loss function

A well-defined loss function is crucial for training. The loss functions of the generator and steganalyzer are presented as follows, separately.

Generative Loss: The generative loss L_G contains three terms: the pixel loss L_p, the contrastive loss L_c, the adversarial loss L_a, i.e., $L_G = L_p + L_c + L_a$. The pixel loss L_p is measured by the ℓ_2 norm of the difference between the original image and its adversarial version to control visual quality, $L_p = \| X^o - X^a \|_2^2$. To learn and preserve sub-regions noise residuals features, the contrastive loss is introduced. The contrastive loss is defined as

$$\begin{aligned} L_c &= (1-y)\frac{1}{2}\| P(G_1(X_l^o)) - P(G_2(X_r^o)) \|_2^2 \\ &+(y)\frac{1}{2}[\max(0, m - \| P(G_1(X_l^o)) - P(G_2(X_r^o)) \|_2)]^2 \end{aligned} \tag{5}$$

Where $m = 0.5$, is the margin, and $P(\cdot)$ represents a global average pooling layer. As for the pairwise label y, $y = 0$ indicates the cover sub-regions, and $y = 1$ indicates the stego sub-regions. Eq. (5) satisfies that the loss is low if the extracted features $P(G_1(X_l^o))$ and $P(G_2(X_r^o))$ are similar.

In addition, we use the adversarial loss [31] to keep the distribution of the adversarial cover before and after embedding as consistent as possible. It can be formulated as

$$L_a = | (Ste(C^a)[0] - Ste(C^a)[1]) - (Ste(S_{C^a})[0] - Ste(S_{C^a})[1]) | \tag{6}$$

where $Ste(\cdot)[0] - Ste(\cdot)[1]$ is the distance between labels.

Steganalysis Loss: As mentioned earlier, steganalysis is usually treated as a binary classification problem. Therefore, the cross-entropy function is used as the steganalysis loss L_S.

$$L_S = log(Ste(C^a)) + log(1 - Ste(S_{C^a})) \tag{7}$$

4. Experiments

In this section, we first explicate the experimental setup. Then, we evaluate the generated adversarial covers in terms of subjective and objective quality. Also, the security is validated from two scenarios. Finally, comparison experiments are conducted.

4.1. Experimental setup

In experiments, three steganographic algorithms, WOW [7], S-UNIWARD [8] and HILL [9], and three steganalyzers, CNN-based XuNet [14] and YeNet [15] and hand-crafted features-based SRM equipped with the ensemble classifier [13] are utilized to verify the misclassification ability of the generated adversarial cover. The employed steganalysis methods all belong to statistical analysis. The statistical analysis method collects a certain number of normal images and corresponding stego images obtained by the publicly available steganographic algorithms as training samples to train a steganalyzer. And the trained steganalyzer is then employed to detect the desired image. The purpose of our model is to generate an enhanced image, which is equivalent to preprocessing the image. Different from the embedding algorithm, the preprocessing method is usually not publicly available, so the steganalyst cannot use the collected image to obtain the corresponding enhanced version, let alone the embedded one of the enhanced version. In this case, the steganalyst only can use the unprocessed images and corresponding embedded ones to train the steganalyzer. Therefore, in our method, the steganalysis is conducted based on the original covers and the corresponding stegos.

We have introduced the latter two steganalyzers earlier. XuNet contains the HPF (high pass filter), five convolutional layers, a fully connected layer, and the Softmax. Concretely, using an ABS (absolute activation) layer after the first convolutional layer, TanH activation function for the first two layers and ReLU for the last three layers, and BN (batch normalization) in each convolutional layer. In our implementation, the image size, learning rate, and other parameters all follow the default settings in these references. The steganalyzers that verify the misclassification ability are trained independently of the proposed model. The experimental dataset is the BOSSBase [40], containing 10000 images with the size of 512×512. Because the size of the input image, adopted by the aforementioned steganalyzers, is 256×256, we resize the BOSSBase dataset to 256×256 to maintain their detection performance. 10000 images are available as covers, and select 5000 images to obtain the corresponding stegos by using the content-adaptive steganographic algorithm, taking the HILL at 0.4 bpp for example. 5000 cover-stego pairs are used for training, and another 5000 covers are used to test. The steganalysis network is pre-trained for 200 epochs on the training set and then iteratively trained for 50 epochs with generator. The pre-training of steganalyzer is to ensure that the constructed adversarial image has better anti-detection ability. Also, during adversarial training, the steganalyzer has strong classification ability at the beginning, which is more conducive to the convergence of

Fig. 3. We observe the visual differences between the generated adversarial cover and its corresponding adversarial stego and the original cover. Here, taking two images as examples, subgraph (a) denotes the original covers, subgraph (b) denotes the adversarial covers, and subgraph (c) denotes the adversarial stegos. Adversarial stegos are obtained by WOW (the top) and S-UNIWARD (the bottom) at 0.4 bpp, respectively.

Table 2
The PSNR and SSIM scores between standard images and the corresponding adversarial ones generated using our method.

Image name	PSNR Score	SSIM Score	Image name	PSNR Score	SSIM Score
Cameraman	47.5691	0.9999	Parrot	49.4038	0.9999
House	47.6794	0.9997	Lena	48.4199	0.9998
Pepper	46.0205	0.9997	Barbara	48.0586	0.9998
Fishstar	48.6940	0.9999	Ship	48.2910	0.9998
Monarch	48.8674	0.9999	Man	46.7103	0.9997
Airplane	48.7088	0.9998	Couple	48.3198	0.9997

Table 3
The PSNR comparisons with the state-of-the-art GAN-based steganographic models with the payload 0.2 and 0.4 bpp. O_W, O_S, and O_H denote the abbreviations of our model combined with three steganographic algorithms WOW, S-UNIWARD, and HILL.

Image name	UT-GAN [21]		CHAT-GAN [26]		O_W		O_S		O_H	
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
Cameraman	60.2620	57.6982	49.2035	48.4486	61.9563	58.1812	**62.5997**	**59.0349**	62.2370	58.8223
House	62.2954	58.7216	53.9682	52.0531	63.0713	59.4248	63.5448	59.7644	**63.5679**	**60.0353**
Pepper	60.9898	58.1926	50.7026	49.6298	62.2370	58.9396	**63.2940**	59.6351	63.2680	**59.7117**
Fishstar	61.1210	58.2720	50.3090	49.2383	62.4808	58.8542	62.9631	59.4950	**63.1950**	**59.6661**
Monarch	61.0986	58.1859	50.2977	49.3878	62.1643	58.4210	62.8419	59.2490	**62.8971**	**59.3277**
Airplane	60.4252	57.6767	49.2073	48.3951	62.1828	58.3431	**62.5134**	**59.0783**	62.3197	58.7080
Parrot	60.5948	57.8608	49.4759	48.7043	61.8536	58.1792	**62.5390**	**58.9141**	61.8209	58.3598
Lena	61.0452	58.1114	53.2180	51.9357	61.9579	58.5380	**63.2121**	**59.5031**	63.0528	59.3487
Barbara	60.7266	57.7794	50.6090	49.7066	62.1845	58.7598	**63.1929**	**59.6117**	62.2251	58.8519
Ship	61.8787	58.5709	52.0098	50.6696	62.4646	58.8519	63.0631	59.5470	**63.4310**	**59.7231**
Man	61.6926	58.0654	51.9844	50.6768	62.7815	58.9380	**63.4176**	**59.6993**	63.1590	59.5608
Couple	61.4812	58.6914	52.0705	50.5759	62.6202	58.9141	**63.3598**	**59.6145**	63.2292	59.5931
Average PSNR	61.1343	58.1522	51.0880	49.9518	62.3296	58.6954	**63.0451**	**59.4289**	62.8669	59.3090

generator. We select the Adamax optimizer with the dynamic learning rate. The initial learning rate is $1e^{-4}$ and adjusted by the MultiStepLR. After training, it only takes less than 5 min to process 5000 test images for verifying visual quality and security using NVIDIA GeForce RTX 2080 Ti. The generated adversarial covers are half for training and half for testing.

Two scenarios are considered, one assuming that the steganalyst is unaware of the adversarial operation, i.e., the adversary-unaware steganalysis. The other is the adversary-aware steganalysis, in which case the steganalyst may use other untargeted steganalyzers or retrain the targeted steganalyzer that has been trained on cover-stego pairs using the adversarial covers and corresponding stegos. It is worth noting that YeNet is the targeted steganalyzer in our experiments.

4.2. Quality evaluation

Fig. 3 illustrates the original covers, the generated adversarial covers, and the homologous adversarial stegos. It is observed that differences among them are not perceived by the human eyes. To objectively evaluate the authenticity of adversarial covers, in contrast with the original covers, we further compute the average PSNR (peak signal to noise ratio) and SSIM (structural similarity) of 5000 adversarial covers. And the average PSNR and SSIM are 47.8267 and 0.9995, respectively. As shown in Table 2, we also adopted twelve 256×256 sized standard images to show in more detail the PSNR and SSIM scores between each image and the corresponding adversarial one generated using our method. Furthermore, Table 3 reports the PSNR comparisons of these standard images before and after embedding secret information by our model and the state-of-the-art GAN-based steganographic models

Table 4
The comparisons of the ability to trick steganalyzers for various methods, using WOW and S-UNIWARD at 0.4 bpp (%).

Method	WOW			S-UNIWARD		
	YeNet	SRM	YeNet (retrain)	YeNet	SRM	YeNet (retrain)
Baseline	79.40	78.44	79.40	75.46	79.66	75.46
Zhang's [30]	50.40	50.40	57.80	50.40	50.30	63.20
Zhou's [31]	49.80	50.10	52.10	49.90	50.00	52.90
Ours	**49.60**	**49.92**	**50.08**	**49.81**	**49.97**	**51.36**

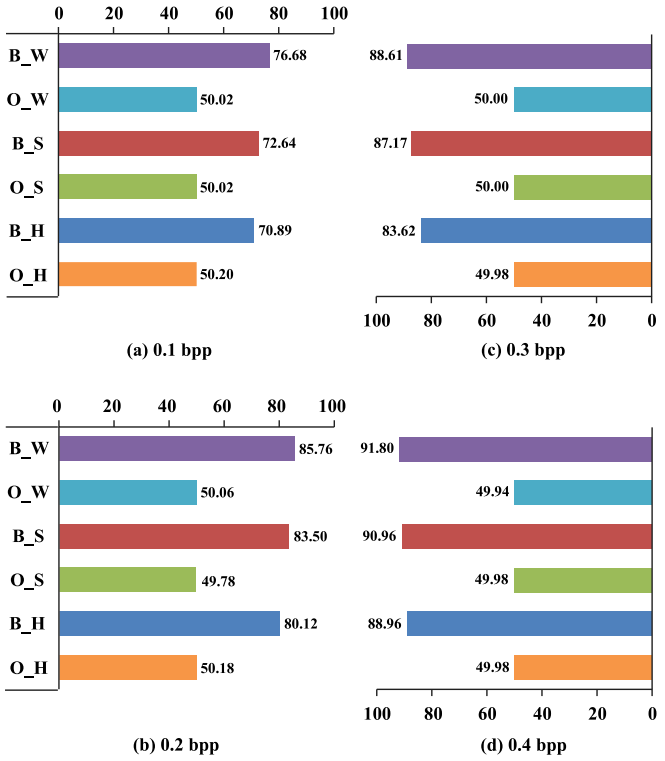

Fig. 4. Performance against the adversary-unaware steganalysis with the payloads from 0.1 to 0.4 bpp. Within each chart, the horizontal axis denotes the classification accuracy (%), and the vertical axis denotes six abbreviations of Baseline and Our model combined with three steganographic algorithms WOW, S-UNIWARD, and HILL. The adopted steganalyzer is YeNet. We can observe that our model can obtain excellent anti-detection performance, and the classification accuracy of the steganalyzer is only around 50%.

[21,26]. All models are trained on the BOSSBase dataset and tested on the standard images. Through observing the objective scores in Tables 2 and 3, we can find that even though the training and test data do not come from the same dataset, our model can still generate high-quality adversarial images. And the corresponding stego images can also achieve the optimal PSNR scores.

4.3. Security analysis

Baseline: The Baseline means the experiments containing embedding and classification are performed on the original covers. The lower the classification accuracy, the higher the security.

Performance Against the Adversary-unaware Steganalysis: In this case, the steganalyst is unaware of the adversarial operation and still uses the targeted steganalyzer YeNet to classify. We conduct experiments on the original cover-stego pairs and the adversarial cover and corresponding stego pairs using WOW, S-UNIWARD, and HILL with the payloads from 0.1 to 0.4 bpp. As described in Fig. 4, experimental results show that the generated adversarial covers can accommodate different steganographic algorithms under the payload of no more than 0.4 bpp. This threshold may be related to the approximately 0.4 bpp of secret information randomly changing the value (0, −1, +1) in proportion during model training as we mentioned earlier. The detection accuracy of steganalyzer sharply reduces to almost random guess, i.e. 50%, without training the specific one according to different payloads and steganographic algorithms.

Performance Against the Adversary-aware Steganalysis: On this occasion, practicable measures taken by the steganalyst include using other untargeted steganalyzers, such as XuNet and SRM, or retraining the targeted steganalyzer YeNet. For the Baseline, it should be noted that there is not much performance difference for steganalyzers using 5000 original cover-stego pairs for training or 7500 original cover-stego pairs for training. Therefore, the same result is employed before and after retraining steganalyzers to save time. Take S-UNIWARD with the payload 0.1 and 0.4 bpp for examples. The experimental results are illustrated in Fig. 5. We can see that the adversarial cover and corresponding stego pairs can still fool the adversary-aware steganalysis. It is further proved that the noise added by the proposed method can not only make steganalyzers lose their classification ability but also make them unable to perceive the existence of the added noise.

4.4. Comparison experiments

The comparison algorithms contain Zhang's [30] and Zhou's methods [31]. Both of them need to design adversarial covers according to different payloads and steganographic algorithms, which undoubtedly increases time overhead. When reproducing the comparison algorithm [31] using the code provided by the author, there is a certain gap between the reproduced performance and the results shown in the published paper, so we choose the published results for comparisons. And for the sake of fairness, we also resize the image to 128 × 128, being consistent with the image size in the comparison algorithms. Considering the limited comparison measures, we only perform the comparison experiments from two scenarios, using WOW and S-UNIWARD at 0.4 bpp. Table 4 reports the ability to trick steganalyzers for various methods. We can see that the performances obtained by these three methods are comparable in most cases. And compared with Zhang's and Zhou's methods, the proposed method can achieve maximum improvements of 11.84% and 2.02% under the condition of retraining the targeted steganalyzer.

Besides, we add the comparisons with the state-of-the-art steganographic methods based on GAN [21,26], further proving the security of the proposed model. Table 5 shows the experimental comparisons of two GAN-based steganographic methods and our method combined with WOW, S-UNIWARD, and HILL under five payloads. SRM, XuNet, and YeNet are employed to evaluate the ability of steganographic methods against steganalysis. To show the effectiveness of the proposed method more intuitively, we also draw the experimental comparisons as Fig. 6. It can be observed that the proposed method can always obtain optimal performance, and the security is significantly improved compared with the GAN-based steganographic approaches.

5. Conclusions

In this work, we present an end-to-end model to quickly generate the visually-pleasing and enough-secure adversarial covers. The

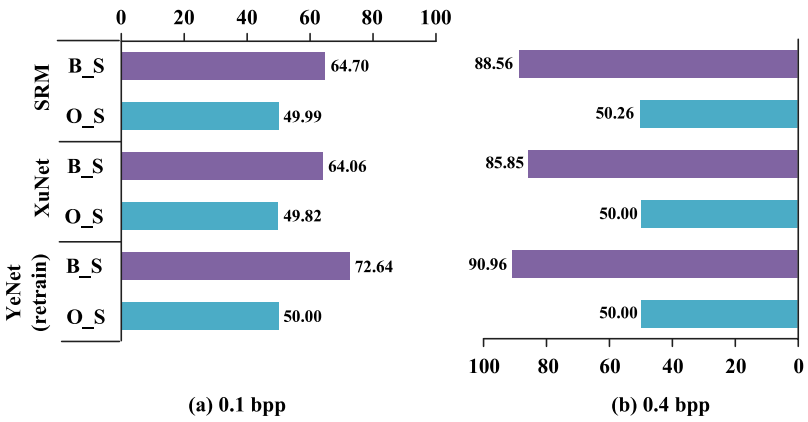

Fig. 5. Performance against the adversary-aware steganalysis with the payload 0.1 and 0.4 bpp. The meaning of the horizontal axis is the same as in Fig. 4, and the vertical axis denotes two abbreviations of Baseline and Our model combined with the steganographic algorithm S-UNIWARD. Because there are three steganalyzers, the vertical axis is duplicated in triplicate. In the scenario, our model shows similar performance as the case of the adversary-unaware steganalysis.

Table 5
The comparisons with the state-of-the-art steganographic methods based on GAN against three steganalyzers, using the payloads from 0.1 to 0.5 bpp. O_W, O_S, and O_H denote the abbreviations of our model combined with three steganographic algorithms WOW, S-UNIWARD, and HILL. The lower the accuracy (%) of the steganalyzer, the stronger the undetectable ability of the steganographic algorithm.

Steganalyzer	Method	Payload				
		0.1 bpp	0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp
SRM	UT-GAN [1]	63.38	72.66	78.92	83.41	87.73
	CHAT-GAN [2]	65.57	74.86	81.13	85.65	88.98
	O_W	**49.99**	**50.00**	50.10	50.02	50.01
	O_S	**49.99**	50.09	50.25	50.26	50.06
	O_H	50.05	**50.00**	**50.00**	**50.00**	**50.00**
XuNet	UT-GAN [1]	57.42	64.48	70.62	73.54	79.06
	CHAT-GAN [2]	62.32	69.28	75.54	78.43	83.68
	O_W	50.08	50.05	50.12	50.06	50.64
	O_S	**49.82**	**50.02**	**50.02**	**50.00**	**46.66**
	O_H	50.00	51.50	50.12	50.12	49.98
YeNet	UT-GAN [1]	64.91	73.81	78.82	79.42	83.70
	CHAT-GAN [2]	69.38	78.26	83.22	85.82	89.08
	O_W	**50.02**	50.06	50.00	**49.94**	50.00
	O_S	**50.02**	**49.78**	50.00	49.98	**49.88**
	O_H	50.20	50.18	**49.98**	49.98	50.38

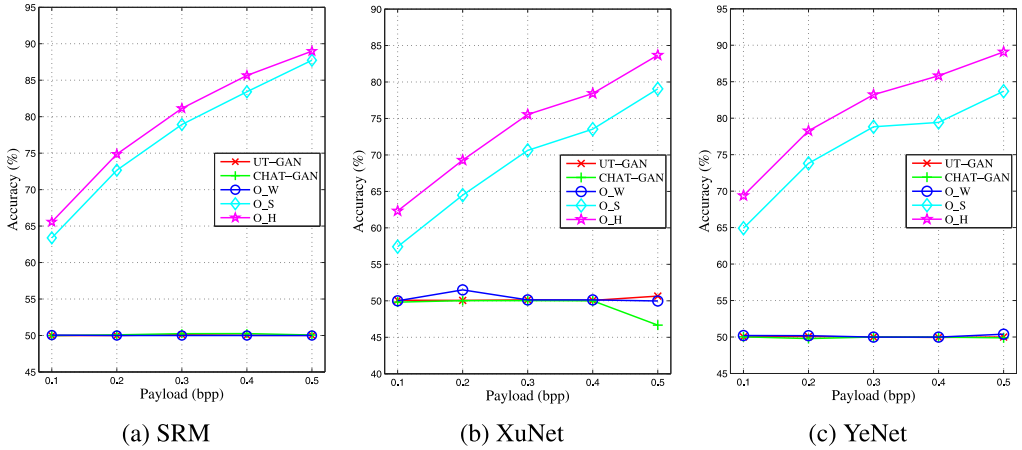

Fig. 6. The comparisons with the state-of-the-art steganographic methods based on GAN against three steganalyzers, using the payloads from 0.1 to 0.5 bpp.

proposed model contains two modules: a generator and a steganalyzer. Specifically, the generator contains the twin block with shared structures and parameters, the fusion block, and convolutional layers. In the twin block, each branch comprises the SRM, a residual block with CBAM, and the convolution–deconvolution group based on U-Net. And a classical CNN-based steganalyzer YeNet is selected as the discriminator to make the adversarial cover possess a more powerful misclassification ability. The results also verify the superiority of our model in terms of quality and security. The ACGIS design can provide some inspiration for future research in steganography and adversarial example. In the future, we will focus on extracting features using more advanced network modules and migrating our work to color images.

CRediT authorship contribution statement

Junxue Yang: Methodology, Conceptualization, Writing,Software, Validation. **Xin Liao:** Supervision, Review, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant Nos. U22A2030, 61972142, 62002112), National Key R&D Program of China (Grant No. 2022YFB3103500), and Hunan Provincial Natural Science Foundation of China (Grant Nos. 2020JJ4212, 2021JJ40117).

References

[1] L. Yang, M. Men, Y. Xue, J. Wen, P. Zhong, Transfer subspace learning based on structure preservation for JPEG image mismatched steganalysis, Signal Process., Image Commun. 90 (2021) 116052.
[2] C. Yang, X. Luo, J. Lu, F. Liu, Extracting hidden messages of MLSB steganography based on optimal stego subset, Sci. China Inf. Sci. 61 (11) (2018) 237–239.
[3] R. Böhme, Advanced Statistical Steganalysis, first ed., in: Information Security and Cryptography, Springer, Berlin, Germany, 2010.
[4] T. Pevny, P. Bas, J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 215–224.
[5] S. Arivazhagan, E. Amrutha, W. Sylvia Lilly Jebarani, Universal steganalysis of spatial content-independent and content-adaptive steganographic algorithms using normalized feature derived from empirical mode decomposed components, Signal Process., Image Commun. 101 (2022) 116567.
[6] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 920–935.
[7] V. Holub, J. Fridrich, Designing steganographic distortion using directional filters, in: Proceedings of IEEE International Workshop on Information Forensics and Security, 2012, pp. 234–239.
[8] V. Holub, J. Fridrich, T. Denemark, Universal distortion function for steganography in an arbitrary domain, EURASIP J. Inform. Secur. 1 (2014) 1–13.
[9] B. Li, M. Wang, J. Huang, X. Li, A new cost function for spatial image steganography, in: Proceedings of IEEE International Conference on Image Processing, 2014, pp. 4206–4210.
[10] B. Li, M. Wang, X. Li, S. Tan, J. Huang, A strategy of clustering modification directions in spatial image steganography, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1905–1917.
[11] M. Hussain, A. Wahab, Y. Idris, A. Ho, K. Jung, Image steganography in spatial domain: A survey, Signal Process., Image Commun. 65 (2018) 46–66.
[12] J. Fridrich, J. Kodovsky, Rich models for steganalysis of digital images, IEEE Trans. Inf. Forensics Secur. 7 (3) (2012) 868–882.
[13] J. Kodovsky, J. Fridrich, V. Holub, Ensemble classifiers for steganalysis of digital media, IEEE Trans. Inf. Forensics Secur. 7 (2) (2012) 432–444.
[14] G. Xu, H. Wu, Y. Shi, Structural design of convolutional neural networks for steganalysis, IEEE Signal Process. Lett. 23 (5) (2016) 708–712.
[15] J. Ye, J. Ni, Y. Yi, Deep learning hierarchical representations for image steganalysis, IEEE Trans. Inf. Forensics Secur. 12 (11) (2017) 2545–2557.
[16] G. Feng, X. Zhang, Y. Ren, Z. Qian, S. Li, Diversity-based cascade filters for JPEG steganalysis, IEEE Trans. Circuits Syst. Video Technol. 30 (2) (2020) 376–386.
[17] D. Volkhonskiy, I. Nazarov, E. Burnaev, Steganographic generative adversarial networks, in: Proceedings of the International Conference on Machine Vision, 2020, pp. 991–1005.
[18] H. Shi, J. Dong, W. Wang, Y. Qian, X. Zhang, SSGAN: Secure steganography based on generative adversarial networks, in: Proceedings of the Pacific Rim Conference on Multimedia, 2017, pp. 534–544.
[19] H. Zi, Q. Zhang, J. Yang, X. Kang, Steganography with convincing normal image from a joint generative adversarial framework, in: Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2018, pp. 526–532.
[20] W. Tang, S. Tan, B. Li, J. Huang, Automatic steganographic distortion learning using a generative adversarial network, IEEE Signal Process. Lett. 24 (10) (2017) 1547–1551.
[21] J. Yang, D. Ruan, J. Huang, X. Kang, Y. Shi, An embedding cost learning framework using GAN, IEEE Trans. Inf. Forensics Secur. 15 (2020) 839–851.
[22] J. Hayes, G. Danezis, Generating steganographic images via adversarial training, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 1954–1963.
[23] J. Zhu, R. Kaplan, J. Johnson, F. Li, HiDDeN: Hiding data with deep networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 657–672.
[24] K.A. Zhang, A. Cuestainfante, L. Xu, K. Veeramachaneni, SteganoGAN: High capacity image steganography with GANs, 2019, arXiv:1901.03892v2, [cs.CV].
[25] C. Yu, Attention based data hiding with generative adversarial networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 1120–1128.
[26] J. Tan, X. Liao, J. Liu, Y. Cao, H. Jiang, Channel attention image steganography with generative adversarial networks, IEEE Trans. Netw. Sci. Eng. 9 (2) (2022) 888–903.
[27] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Proceedings of International Conference on Learning Representations, 2015.
[28] M. Duan, K. Li, L. Xie, Q. Tian, B. Xiao, Towards multiple black-boxes attack via adversarial example generation network, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 264–272.
[29] F. Li, Z. Yu, C. Qin, GAN-based spatial image steganography with cross feedback mechanism, Signal Process. 190 (2022) 108341.
[30] Y. Zhang, W. Zhang, K. Chen, J. Liu, Y. Liu, N. Yu, Adversarial examples against deep neural network based steganalysis, in: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, 2018, pp. 67–72.
[31] L. Zhou, G. Feng, L. Shen, X. Zhang, On security enhancement of steganography via generative adversarial image, IEEE Signal Process. Lett. 27 (2020) 166–170.
[32] W. Tang, B. Li, S. Tan, M. Barni, J. Huang, CNN-based adversarial embedding for image steganography, IEEE Trans. Inf. Forensics Secur. 14 (8) (2019) 2074–2087.
[33] S. Bernard, T. Pevny, 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] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2015, arXiv:1511.06434v2 [cs.LG].
[35] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 214–223.
[36] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 2391–2400.
[37] K. Fu, D. Fan, G. Ji, Q. Zhao, J. Shen, C. Zhu, Siamese network for RGB-D salient object detection and beyond, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2022) 5541–5559, http://dx.doi.org/10.1109/TPAMI.2021.3073689.
[38] S. Woo, J. Park, J. Lee, I. Kweon, CBAM: Convolutional block attention module, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 3–19.
[39] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Proceedings of the Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[40] P. Bas, T. Filler, T. Pevny, Break our steganographic system: The ins and outs of organizing boss, in: Proceedings of International Workshop on Information Hiding, 2011, pp. 59–70.