mathematics

MDPI

Article

Image Steganography and Style Transformation Based on Generative Adversarial Network

Li Li [1], Xinpeng Zhang [1,*], Kejiang Chen [2], Guorui Feng [1], Deyang Wu [1] and Weiming Zhang [2]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; llichn@shu.edu.cn (L.L.); grfeng@shu.edu.cn (G.F.); wdyang@shu.edu.cn (D.W.)
[2] CAS Key Laboratory of Electro-Magnetic Space Information, University of Science and Technology of China, Hefei 230027, China; chenkj@mail.ustc.edu.cn (K.C.); zhangwm@ustc.edu.cn (W.Z.)
* Correspondence: xzhang@shu.edu.cn

Abstract: Traditional image steganography conceals secret messages in unprocessed natural images by modifying the pixel value, causing the obtained stego to be different from the original image in terms of the statistical distribution; thereby, it can be detected by a well-trained classifier for steganalysis. To ensure the steganography is imperceptible and in line with the trend of art images produced by Artificial-Intelligence-Generated Content (AIGC) becoming popular on social networks, this paper proposes to embed hidden information throughout the process of the generation of an art-style image by designing an image-style-transformation neural network with a steganography function. The proposed scheme takes a content image, an art-style image, and messages to be embedded as inputs, processing them with an encoder–decoder model, and finally, generates a styled image containing the secret messages at the same time. An adversarial training technique was applied to enhance the imperceptibility of the generated art-style stego image from plain-style-transferred images. The lack of the original cover image makes it difficult for the opponent learning steganalyzer to identify the stego. The proposed approach can successfully withstand existing steganalysis techniques and attain the embedding capacity of three bits per pixel for a color image, according to the experimental results.

Keywords: Generative Adversarial Network (GAN); image steganography; style transfer

MSC: 68T07

Citation: Li, L.; Zhang, X.; Chen, K.; Feng, G.; Wu, D.; Zhang, W. Image Steganography and Style Transformation Based on Generative Adversarial Network. *Mathematics* **2024**, *12*, 615. https://doi.org/10.3390/math12040615

Academic Editors: Guangwei Gao, Juncheng Li and Zhi Li

Received: 9 January 2024
Revised: 1 February 2024
Accepted: 3 February 2024
Published: 19 February 2024

1. Introduction

Image steganography is a concealed communication method that uses seemingly benign digital images to conceal sensitive information. An image with hidden messages is known as a stego. The existing mainstream approaches for image steganography are content-adaptive, which embed secrets into highly textured or noisy regions by minimizing a heuristically defined distortion function, which measures the statistical detectability or distortion. Based on the near-optimal steganographic coding scheme [1,2], numerous efficient steganographic cost functions have been put forth over the years, and many of them are based on statistical models [3,4] or heuristic principles [5–7]. The performance of steganography could also be enhanced by taking into account the correlations between nearby picture elements, such as in [8–10].

Image steganalysis, on the other hand, seeks to identify the presence of a hidden message inside an image. Traditional steganalysis methods are based on statistical analysis or training a classifier [11] based on hand-crafted features [12–14]. In recent years, deep neural networks have been proposed for steganalysis [15–19], and they have outperformed traditional methods, which challenges the security of the steganography. To defend against steganalysis, some researchers have proposed embedding secret messages using deep neural networks and simulating the competitionbetween steganography and steganalysis by a Generative Adversarial Network (GAN), which alternatively updates a generator

and a discriminator by which enhanced cover images or distortion costs can be learned. However, since these methods embed messages based on an existing image, it is possible for the adversary to generate cover–stego pairs, which will provide more information for the steganalysis. To solve this problem, some works utilize GANs to learn how to map the pieces of the secret information to the stego and directly produce stego images without the cover [20–25]. But, the images obtained by the GAN are not satisfying in terms of the visual quality due to the difficulty of the image-generation task.

The goal of the above-mentioned methods is to keep the stego images indistinguishable from the unprocessed natural images since the transfer of the natural images has been a common phenomenon in recent years. Recently, with the rapid growth of AGI, the well-performing image-generation and image-processing models have emerged in great numbers, such as dalle2 [26] and stable diffusion [27], increasing the attention to the steganography of the AI-generated or -processed images [28,29]. Among the images produced by AI, the art-style images have become more popular on social networks, thereby generating stegos that are indistinguishable from style-transferred images, which could be a new way for high capacity and secure steganography. In [30], Zhong et al. proposed a steganography method in stylized images. They produced two similar stylized images with different parameters, and one of them was used for embedding and the other as a reference. However, because it remains dependent on the framework of embedding distortion and STC coding, the adversary may detect the stego by generating cover–stego pairs and training a classifier; thereby, the stego images face the risk of being detected. In this paper, we propose to encode secret messages into images at the same time as the generation of style-transferred images. The contributions of the paper are as below:

1. We designed a framework for image steganography during the process of image style transfer. The proposed method is more secure compared to traditional steganography since the steganalysis without the corresponding cover–stego pairs is difficult.
2. We validated the effectiveness of the proposed method by experiments. The results showed that the proposed approach can successfully embed 1 bpcpp, and the generated stego cannot be distinguished from the clean style-transferred images generated by a model without steganography. The accuracy of the recovered information was 99%. Though it was not 100%, this can be solved by coding secret information using error-correction codes before hiding them in the image.

2. Related Works

2.1. Image Steganography

The research on steganography is usually based on the "prisoner's problem" model, which was proposed by American scholar Simmons in 1983 and is described as follows: "Assuming Alice and Bob are held in different prisons and wish to communicate with each other to plan their escape, but all communication must be checked by the warden Wendy". The steganographic communication process is shown in Figure 1. The sender, Alice, hides the message in a seemingly normal carrier by selecting a carrier that Wendy allows and using the key shared with the receiver, Bob. This process can be represented as:

$$Emb(c, m, k) = s \tag{1}$$

Then, the carrier is transmitted to the receiver, Bob, through a public channel. Bob receives the carrier containing the message and uses a shared key to extract the message:

$$Ext(s, k) = m. \tag{2}$$

Wendy, the monitor in the public channel, aims to detect the presence of covert communication.

Existing steganography methods can be divided into three categories: (1) cost-based steganography, (2) model-based steganography, and (3) generative steganography.

Figure 1. Comparison with traditional image steganography and style transfer steganography.

2.1.1. Cost-Based Steganography

Each cover element $i \in 1, \cdots N$ of the cover is allocated a cost $\rho_i \geq 0$ and a probability β_i for modifying its value according to the image content, and techniques such as UNIWARD, WOW, and HILL propose a variety of cost-designing methods. The objective of cost-based steganography is to embed the secret message into the cover in a way that minimizes the sum of the predicted costs of all modified pixels, which is calculated by $d = \sum_{i=1}^{N} \beta_i \rho_i$. To this end, the problem of secret embedding is recognized as source coding with a fidelity constraint, and near-optimal coding schemes Syndrome-Trellis Codes (STCs) and Steganographic Polar Codes (SPCs) have been developed. Cost-based steganography adaptively embeds secrets; hence, the steganography is imperceptible. However, the costs are occasionally determined via heuristic methods and cannot be mathematically associated with the potential of the changes in the embedding being detected. Moreover, when a well-informed adversary is cognizant of the changing rates of the embedding, which is taken as a kind of side information in steganalysis, it can be used by the adversary to improve the steganalysis's accuracy by utilizing selection-channel-aware features or Convolutional Neural Networks.

2.1.2. Model-Based Steganography

Model-based steganography establishes a mathematical model for the distribution of carriers, aiming to embed messages while preserving the inherent distribution model as much as possible. MiPOD is an example of the model-based steganographic scheme. It assumes the noise residuals in a digital image follow a Gaussian distribution with zero mean and variances σ_i^2, which are estimated for each cover element i. The messages are embedded, aiming to reduce the effectiveness of the most-advanced detector that an adversary can create. While this approach is theoretically secure, challenges arise due to variations in the distribution models among multimedia data, such as images and videos, acquired by different sensors. Furthermore, the influence of distinct temporal and environmental factors on the pixel distribution complicates the identification of a universally applicable model for accurately fitting real-world distributions.

2.1.3. Coverless Steganography

Unlike cost-based methods or model-based methods, where a prepared cover object is used to hide the data by modifying the pixel values, coverless steganography is based on the principle that natural carriers may carry the secret information that both parties want to transmit in secret communication. This does not require preparing the cover to be modified, but aims to embed information directly within the carrier medium itself, without relying on modifying a distinct cover. Traditional methods achieve this by selecting an image that is suitable for the message to be transmitted. With the development of generative models, recent research has proposed to embed the messages during the image-generation or processing.

2.2. Image Style Transfer

Image style conversion methods can be divided into non-realistic rendering (Non) Photorealistic Rendering (NPR) methods and computer vision methods. The NPR methods have developed into an important area in the field of computer graphics; however, most NPR stylization algorithms are designed for specific artistic styles, and it is not easily extended to other styles. The method of computer vision regards style transformation as a texture synthesis problem, that is the extraction and transformation from the source texture to the target texture. The framework of "image analogy" learning achieves universal style conversion by learning from examples of the provided unshaped and stereotypical images. But, these methods only use low-level image patterns.

Physical features cannot effectively provide advanced image structural features. Inspired by Convolutional Neural Networks (CNN)s, Gatys et al. [31] first studied how to use Convolutional Neural Networks to transform natural images into famous painting styles, such as van Gogh's Starry Night. They proposed modeling the content of photos as intermediate-layer features of pretrained CNNs and modeling artistic styles as the statistics of intermediate-layer features.

With the rapid development of style transition networks based on CNNs, the efficiency of image style conversion has gradually improved, and image-processing software such as Prisma and Deep Forger have become popular, making sending artistic style images on social platforms a common phenomenon. Therefore, covert communication using stylized images as carriers should not be easily suspected. Based on this, this section proposes a steganography method for image style transfer, which embeds secret messages while performing image stylization, making the generated encrypted image indistinguishable from the clean stylized image, improving the steganography's security and capacity.

3. Proposed Methods

It is shown that deep neural networks can learn to encode a wealth of relevant information by invisible perturbations [24]. Image style transfer could be taken as encoding the target style information into the content image. Therefore, we encoded the secret information during the process of image style transfer, directly creating a stylized image with hidden secret messages, as opposed to first computing the steganographyand then applying encoding methods to the image. The style-transferred image containing the secret message is expected to be indistinguishable from the one without the secret message, and to enhance its visual quality, a GAN model was used, where SRNet was utilized as the discriminator, which learns the detailed features of the image and performs well at distinguishing the traditional stego and cover.

As shown in Figure 2, the network architecture consists of four parts: (1) a generator G, which takes the content image and the to-be-embedded message as the inputs, simultaneously achieving style transformation and information embedding; (2) a message extractor E, which is trained along with G, takes the stego image as the input, and precisely retrieves hidden information; (3) a discriminator A, which is iteratively updated with the generator and extractor; and (4) a style transformer loss-computing network L, which is a pretrained VGG model; it is employed to determine the resulting image's style and content loss. The whole model is trained by the sender, and when the model is well trained, the message-extraction network E is shared with the receiver to extract the secret messages that are hidden in the received image. The implementation details of each part are as follows.

In our implementation, we adopted the architecture of image transformation networks in [32] as the generator G; it first utilizes two stride-2-convolutions to down-sample the input, followed by several residual blocks, then two convolutional layers with stride $1/2$ are used to upsample, followed by a stride-1 convolutional layer, which uses a 9×9 kernel. Instance Normalization [33] is added to the start and end of the network.

To encode secret messages during the image style transfer, we concatenated the message M of size $C_m \times H \times W$ with the output of the first convolutional layer with respect to the input content image X_c of size $C \times H \times W$ and took the resulting tensor as the input

of the second convolutional layer; in this way, we obtain a feature map that contains both the secret messages and the input content. The following architecture is like an encoder–decoder, which first combines and condenses the information and, then, restores an image with the condensed feature. The final output of G is a style-transferred image Y_s of size $C * H * W$, which also contains the secret messages. The details of the architecture are shown in Table 1.

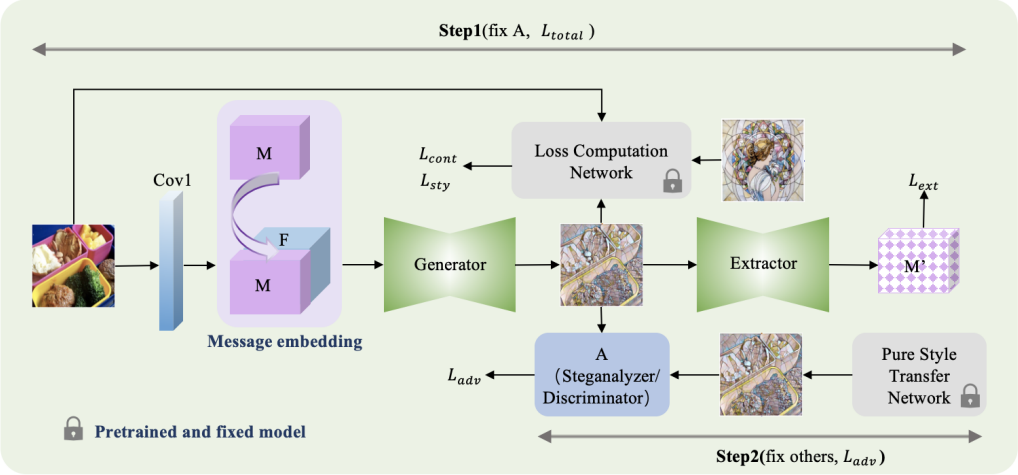

Figure 2. Framework of hiding information in style transform network.

Table 1. Structure of message-embedding network and message-extraction network.

Message-Embedding Network		Message-Embedding Network	
Network Layer	**Output Size**	**Network Layer**	**Output Size**
input	$3 \times 256 \times 256$	input	$3 \times 256 \times 256$
padding(40×40)	$3 \times 336 \times 336$	$3 \times 9 \times 9$ conv, step 1	$3 \times 256 \times 256$
$32 \times 9 \times 9$ conv, step 1	$32 \times 336 \times 336$	$32 \times 3 \times 3$ conv, step 1/2	$32 \times 128 \times 128$
secret message	$3 \times 336 \times 336$	$64 \times 3 \times 3$ conv, step 1	$64 \times 64 \times 64$
message concat	$35 \times 336 \times 336$	residual block, 128 filters	$128 \times 64 \times 64$
$64 \times 3 \times 3$ conv, step 2	$64 \times 168 \times 168$	residual block, 128 filters	$128 \times 68 \times 68$
$128 \times 3 \times 3$ conv, step 2	$128 \times 84 \times 84$	residual block, 128 filters	$128 \times 72 \times 72$
residual block, 128 filters	$128 \times 80 \times 80$	residual block, 128 filters	$128 \times 76 \times 76$
residual block, 128 filters	$128 \times 76 \times 76$	residual block, 128 filters	$128 \times 80 \times 80$
residual block, 128 filters	$128 \times 72 \times 72$	$128 \times 3 \times 3$ conv, step 2	$128 \times 84 \times 84$
residual block, 128 filters	$128 \times 68 \times 68$	$64 \times 3 \times 3$ conv, step 2	$64 \times 168 \times 168$
residual block, 128 filters	$128 \times 64 \times 64$	$32 \times 9 \times 9$ conv, step 2	$32 \times 336 \times 336$
$64 \times 3 \times 3$ conv, step 1/2	$64 \times 128 \times 128$	$3 \times 9 \times 9$ conv, step 1	$3 \times 336 \times 336$
$32 \times 3 \times 3$ conv, step 1/2	$32 \times 256 \times 256$		
$3 \times 9 \times 9$ conv, step 1	$3 \times 256 \times 256$		

3.1. Style Transfer Loss Computing

The resulting images should possess similar content to X_c and possess the target style, which is defined by a target style image X_s. For this reason, we applied a loss calculation network L to quantify in the high-level content the difference between the resulting image and X_c and style difference between the resulting image and X_s, respectively. L is implemented as a 16-layer VGG network, which is pre-trained on the ImageNet dataset for the image-classification task in advance. To achieve style transfer, two perceptual losses were designed, namely the content reconstruction loss and style reconstruction loss.

3.1.1. Content Reconstruction Loss

We define the content reconstruction loss as the difference between the activations of the intermediate layers of L with respect to X_c and Y_s as the inputs. The activation maps of

the j-th layer of the network in terms of the input image x are represented as $\phi_j(x)$, then the content loss is defined as the mean-squared error between the activation map of Y_s and X_c, represented as:

$$L_{\text{cont}}(X_c, Y_s, j) = \frac{1}{C_j H_j W_j} \sum_{i,j} \|\phi(X_c) - \phi(Y_s)\|_2 \tag{3}$$

It is shown in [32] that the high-level content of the image is kept in the responses of the higher layers of the network, while the detailed pixel information is kept in the responses of the lower layers. Therefore, we calculated the perceptual loss for style transfer at the high layers. This does not require that the output image Y_s perfectly matches X_c; instead, it encourages it to be perceptually similar to X_c; hence, there is extra room for us to implement style transfer and steganography.

3.1.2. Style Reconstruction Loss

To implement style transfer, except for content loss, style reconstruction loss is also required to penalize the differences in style such as the colors and textures between Y_s and X_s when Y_s deviates from the input X_c in terms of style. To this end, we first define the Gram matrix $G_j^{\phi}(x)$ to be a matrix of size $C_j \times C_j$, and the elements of $G_j^{\phi}(x)$ are defined as:

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j \times H_j \times W_j} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \tag{4}$$

To achieve better performance, we calculated the style loss L_{sty} from a set of layers J instead of a single layer j. Specifically, L_{sty} is defined as the sum of the losses for each layer $j \in J$, as described in Equation (5).

$$L_{\text{sty}} = \sum_{j \in J} \left\| G_j^{\phi}(X_{\text{sty}}) - G_j^{\phi}(Y_s) \right\|_2 \tag{5}$$

3.2. Extractor

To accurately recover the embedded information, a message-extraction network E, which has the same architecture as the generator G, is trained together with G. It takes the generated image, i.e., Y_s, as the input and outputs O of size $C_m \times H \times W$. The revealed message M' is obtained according to O:

$$M'_{i,j,k} = \begin{cases} 0 & \text{if} \quad O_{i,j,k} < 0 \\ 1 & \text{if} \quad O_{i,j,k} \geq 0 \end{cases} \tag{6}$$

The loss for revealing the information is defined as the mean-squared error between the embedded message M and the extracted message M':

$$L_{\text{ext}} = \left\| M - M' \right\|_2 \tag{7}$$

When the model is well trained, E is shared between Alice and Bob for convert communication, which plays the role of the secret key. Therefore, it is crucial to keep the secret of the trained E.

3.3. Adversary

To enhance the resulting Y_s's visual quality, an adversarial training technique is applied, where SRNet [18] is applied as a discriminator to classify the generated style-transferred images containing secret messages and clean style-transferredimages generated

by a style-transfer network without the steganography function. The cross-entropy loss is applied to measure the performance of the discriminator, which is defined as Equation (8).

$$L_{\text{adv}} = y\log\phi(x) + (1-y)\log(1-\phi(x)) \tag{8}$$

When updating the generator, the objective is to maximize L_{adv}, while when updating the discriminator, the objective is to minimize L_{adv}.

3.4. Training

In the training process, we iteratively update the parameters of the generator and adversary. Each iteration contains two epochs: in the first epoch, we leave the parameters of the discriminator unchanged and update the parameters of the first convolution layer, the generator, and the extractor by minimizing the content loss L_{cont}, style loss L_{sty}, and message extraction loss L_{ext}, but maximizing the discriminator's loss L_{adv}; hence, the total loss for training is defined as follows:

$$L_{total} = \alpha L_{\text{cont}} + \beta L_{\text{sty}} + \lambda L_{\text{ext}} - \gamma L_{\text{adv}}, \tag{9}$$

where α, β, λ, and γ are hyper-parameters to balance the content, style, message-extraction accuracy, and risk of being detected by the discriminator. In the second epoch, we update the parameters of the adversary by using the loss defined in Equation (8) while keeping the remaining parameters fixed.

4. Experiments

To verify the efficiency of the suggested approach, we randomly chose a style image from the WikiArt dataset as the target style and randomly took 20,000 content images from COCO [34], 10,000 for training and 10,000 for testing. We repeated the experiments 10 times. All the images were resized to 512×512 px with the channel of 3, and the messages to be embedded were binary data with the size of $3 \times 512 \times 512$, i.e., the payload was set as 1 bit per channel per pixel (bpcpp). In the training, the Adam optimizer was applied, and the learning rate was set as 1×10^{-4}. We trained the network for 200 epochs. The performance of the proposed method was evaluated from two aspects: (1) the accuracy rate of message extraction and (2) the ability to resist steganalysis. To demonstrate the versatility and robustness of the proposed method, we also validated the proposed method on the other style image on the Internet.

4.1. Message Extraction Accuracy Analysis

We assumed the sender and the receiver share the parameters and architecture of the extractor, the adversary knows the algorithm for data hiding and can train a model by herself but will obtain mismatched parameters. We explored, in such a situation, whether the hidden message can be extracted accurately by the receiver and whether the secret messages could be leaked to the adversary.

We trained five models of the same architecture, but with different random seeds, and these architectures are illustrated in Figure 2. The well-trained networks are represented as $\text{Net}_1, \text{Net}_2, \text{Net}_3, \text{Net}_4$, and Net_5, respectively. We randomly split the content dataset into two separate sets, one for testing and the other for training. The secret messages to be embedded were randomly generated binary sequences and were reshaped as $3 \times 256 \times 256$. In the testing stage, we extracted the hidden messages by using extractors from different trained models. The results are displayed in Table 2, from which we can infer that the matched extractor can successfully extract the concealed message, and the accuracy rate of the extracted message reached 99.2%, demonstrating that the receiver could accurately recover the messages. But, an adversary cannot steal the secret messages hidden by the proposed method since the mismatched extractor can only recover less than 50% of the messages.

Table 2. Message recovery accuracy using different extractors.

$\text{Net}_{\text{train}}$ \ Net_{test}	Net_1	Net_2	Net_3	Net_4	Net_5
Net_1	**0.99**	0.39	0.31	0.28	0.32
Net_2	0.37	**0.99**	0.28	0.23	0.38
Net_3	0.31	0.19	**0.99**	0.33	0.41
Net_4	0.40	0.29	0.31	**0.99**	0.34
Net_5	0.29	0.32	0.37	0.19	**0.98**

The results of using the matched extractor is represented in bold font. Net_1, Net_2, Net_3, Net_4, and Net_5 are the same architectures as illustrated in Figure 2.

4.2. Security in Resisting Steganalysis

To verify the security of the embedded secret messages, we compared the generated stego style-transferred images with the clean style-transferred images generated by the style-transfer network without steganography [32]. We trained four networks M_{c1}, M_{c2}, M_{s1}, and M_{s2}. M_{s1} and M_{s2} are the same architecture proposed in this paper, but with different parameters; M_{c1} and M_{c2} are style-transfer networks without steganography [32]. The generated images are displayed in Figure 3, where it is clear that the message embedding had no effect on the image visually.

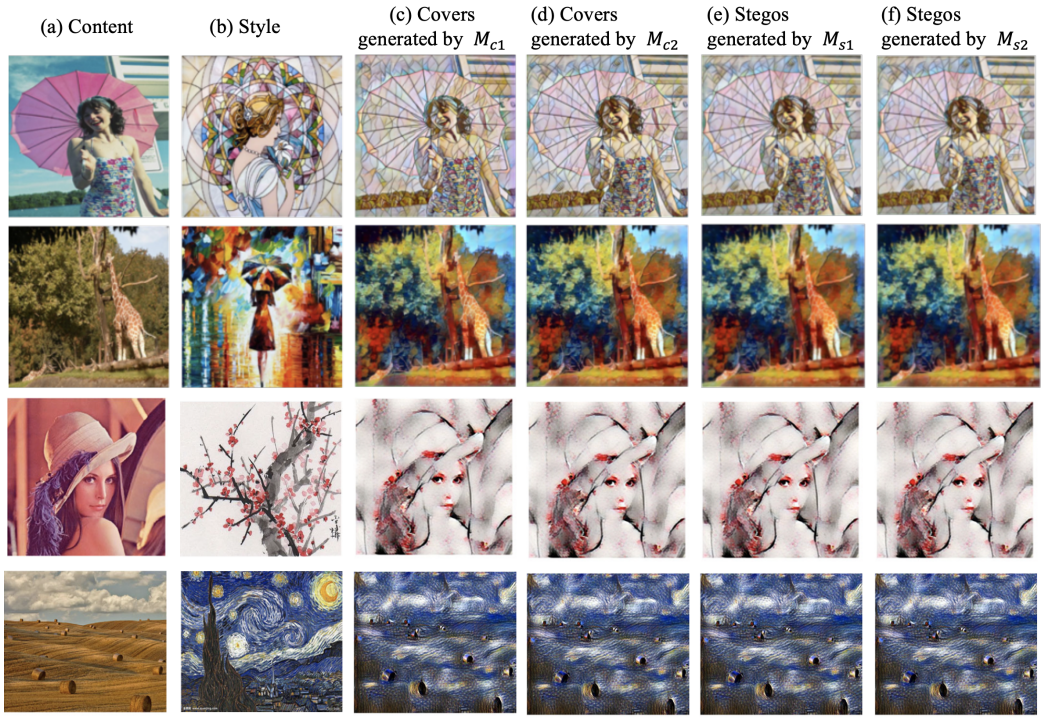

Figure 3. Comparison of clean style-transferred images without steganography (columns (**c**,**d**)) and stego style-transferred images (columns (**e**,**f**)).

The residual of clean image and stego image with secret are shown in Figure 4. It should be noted that the difference between the generated stegos style-transferred and style-transferred images without hidden messages is not only caused by the message embedding, but also due to the different parameters of the model, e.g., the images generated by M_1 are different from those by M_2, but are also different from M_3 and M_4. Thereby, it is difficult to tell whether the image has been produced by a style-transfer network with the steganography function or by another ordinary style-transfer network without steganography. To verify the security of the proposed method, we assumed the attacker is trying to distinguish the generated stego from the cover generated by other normal style-transfer networks without the steganography function. According to the Kerckhoff

principle, we considered a powerful steganalyzer who knows the target style image and all the knowledge of the model (i.e., the architecture and parameters) the steganographer has used. In this case, the attacker can generate the same stego as the steganographer, taking the generated stegos as positive samples and the covers generated by the models as negative samples to train a binary classifier. We applied different steganalysis methods, including using traditional SPAM [13] and SRM [14] features to train a classifier, as well as using the deep learning methods XuNet [16] and SRNet [18]. Similar to steganalysis, we preserved the cover and stego of the same content in the same batch when training the deep-learning-based steganalyzer. Table 3 contains the experimental findings. The average testing errors were all about 0.5, confirming the safety of the suggested procedure. We compared the security of the proposed method with other state-of-the-art steganography methods. The performance under a 0.4 bpp payload is shown in Table 4. It can be seen that the detection error of our method was about 0.5, which equals random guessing; hence, we can infer that our method cannot be detected. Since traditional methods embed secrets by modifying the pixel value of the original image, the modification traces could be reflected by some statistical features or be learned by a deep neural network. Instead, the proposed method embeds the secrets during the image generation; there is no exact cover for the steganalyzer to refer to, so it is difficult to detect.

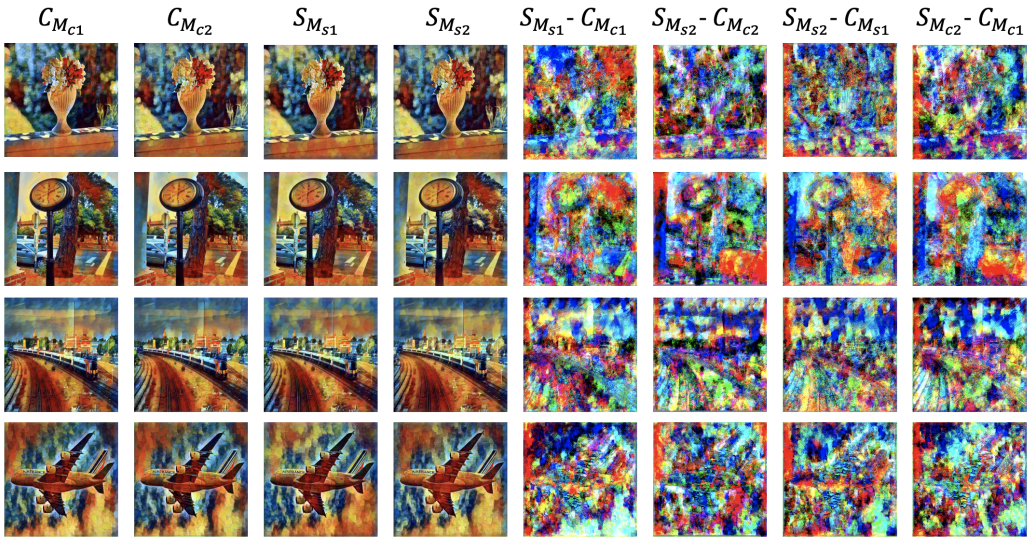

Figure 4. Residual of the style transferred image with and without secret information: $C_{M_{c1}}$, $C_{M_{c2}}$ respectively referred to the style transferred image generated by the clean model M_{c1} and M_{c2}, $S_{M_{s1}}$, $S_{M_{s2}}$ respectively referred to the style transferred image with secret message generated by the steganography model M_{s1} and M_{s2}.

Table 3. Average error of stego with 1 bpp under the detection of different steganalysis methods.

Steganalysis Method	SPAM [13]	SRM [14]	XuNet [16]	SRNet [18]	SCA-SRNet [18]
P_E	0.48	0.49	0.51	0.47	0.48

Table 4. Detection error comparison with different steganography methods with 0.4 bpp.

Steganography	WOW [5]	SUNIWAR [6]	HILL [7]	Ours
SRNet	0.91	0.89	0.86	0.47
SCA-SRNet	0.92	0.91	0.87	0.49

5. Conclusions

In this study, we proposed a high-capacity and safe method for image steganography. We hid secret messages in an art-style image in the process of image generation by a GAN model. It was verified by experiments that the proposed approach can achieve a high capacity of 1 bpcpp, and the generated images cannot be distinguished from the clean images of the same content and style. The proposed method provides a new way for covert communication on social networks. However, there are still some limitations in its application. The message recovery accuracy did not achieve 100%; in addition, there will be complex noise in the real-world communication channel, and some platforms will compress the image before transmission, which will decrease the accuracy of message recovery. In the future, we will consider performing error-correction coding on secret messages before embedding them into the image and explore how to improve the robustness of the steganography.

Author Contributions: Conceptualization, L.L.; Methodology, L.L. and K.C.; Software, L.L.; Validation, K.C.; Writing—original draft, L.L.; Writing—review & editing, G.F. and D.W.; Visualization, D.W.; Supervision, X.Z. and W.Z.; Project administration, X.Z. and G.F.; Funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China under Grants U22B2047, U1936214, and 62302286 and the China Postdoctoral Science Foundation under Grant No. 2023M742207.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Filler, T.; Judas, J.; Fridrich, J. Minimizing Additive Distortion in Steganography using Syndrome-Trellis Codes. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 920–935. [CrossRef]
2. Yao, Q.; Zhang, W.; Chen, K.; Yu, N. LDGM Codes Based Near-optimal Coding for Adaptive Steganography. *IEEE Trans. Commun.* **2023**, *2023*, 1. [CrossRef]
3. Pevný, T.; Filler, T.; Bas, P. Using high-dimensional image modelsto perform highly undetectable steganography. In Proceedings of the International Workshop on Information Hiding, Calgary, AB, Canada, 28–30 June 2010; pp. 161–177.
4. Sedighi, V.; Cogranne, R.; Fridrich, J. Content-Adaptive Steganography by Minimizing Statistical Detectability. *IEEE Trans. Inf. Forensics Secur.* **2015**, *11*, 221–234. [CrossRef]
5. Holub, V.; Fridrich, J. Designing Steganographic Distortion Using Directional Filters. In Proceedings of the IEEE Workshop on Information Forensic and Security (WIFS), Tenerife, Spain, 2–5 December 2012; pp. 234–239.
6. Holub, V.; Fridrich, J.; Denemark, T. Universal Distortion Function for Steganography in an Arbitrary Domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1–13. [CrossRef]
7. Li, B.; Wang, M.; Huang, J.; Li, X. A new cost function for spatial image steganography. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 4206–4210.
8. Li, B.; Wang, M.; Li, X.; Tan, S.; Huang, J. A strategy of clustering modification directions in spatial image steganography. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1905–1917.
9. Denemark, T.; Fridrich, J. Improving steganographic security by synchronizing the selection channel. In Proceedings of the 3rd ACM Information Hiding and Multimedia Security Workshop, Portland, OR, USA, 17–19 June 2015; pp. 5–14.
10. Li, W.; Zhang, W.; Chen, K.; Zhou, W.; Yu, N. Defining joint distortion for JPEG steganography. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 5–16.
11. Kodovský, J.; Fridrich, J.; Holub, V. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 432–444. [CrossRef]
12. Holub, V.; Fridrich, J. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 219–228. [CrossRef]
13. Li, B.; Li, Z.; Zhou, S.; Tan, S.; Zhang, X. New steganalytic features for spatial image steganography based on derivative filters and threshold LBP operator. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1242–1257. [CrossRef]
14. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [CrossRef]
15. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Deep learning for steganalysis via Convolutional Neural Networks. *Proc. SPIE* **2015**, *9409*, 94090J.

16. Xu, G.; Wu, H.Z.; Shi, Y.Q. Structural design of Convolutional Neural Networks for steganalysis. *IEEE Signal Process. Lett.* **2016**, *23*, 708–712. [CrossRef]
17. Ye, J.; Ni, J.; Yi, Y. Deep learning hierarchical representations for image steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2545–2557. [CrossRef]
18. Boroumand, M.; Chen, M.; Fridrich, J. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1181–1193. [CrossRef]
19. Butora, J.; Yousfi, Y.; Fridrich, J. How to Pretrain for Steganalysis. In Proceedings of the 9th Information Hiding and Multimedia Security Workshop, Brussels, Belgium, 22–25 June 2021.
20. Zhang, J.; Chen, K.; Li, W.; Zhang, W.; Yu, N. Steganography with Generated Images: Leveraging Volatility to Enhance Security. *IEEE Trans. Dependable Secur. Comput.* **2023**, *2023*, 1–12. [CrossRef]
21. Chen, K.; Zhou, H.; Wang, Y.; Li, M.; Zhang, W.; Yu, N. Cover Reproducible Steganography via Deep Generative Models. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 3787–3798. [CrossRef]
22. Zhu, J.; Kaplan, R.; Johnson, J.; Li, F.F. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 657–672.
23. Tan, J.; Liao, X.; Liu, J.; Cao, Y.; Jiang, H. Channel Attention Image Steganography with Generative Adversarial Networks. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 888–903. [CrossRef]
24. Tang, W.; Li, B.; Mauro, B.; Li, J.; Huang, J. An automatic cost learning framework for image steganography using deep reinforcement learning. *IEEE Trans. Inf. Forensics Secur.* **2020**, *16*, 952–967. [CrossRef]
25. Guan, Z.; Jing, J.; Deng, X.; Xu, M.; Jiang, L.; Zhang, Z.; Li, Y. DeepMIH: Deep invertible network for multiple image hiding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 372–390. [CrossRef]
26. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv* **2022**, arXiv:2204.06125.
27. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
28. Bui, T.; Agarwal, S.; Yu, N.; Collomosse, J. RoSteALS: Robust Steganography using Autoencoder Latent Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 933–942.
29. Yu, J.; Zhang, X.; Xu, Y.; Zhang, J. CRoSS: Diffusion Model Makes Controllable, Robust and Secure Image Steganography. *arXiv* **2023**, arXiv:2305.16936.
30. Zhong, N.; Qian, Z.; Wang, Z.; Zhang, X. Steganography in stylized images. *J. Electron. Imaging* **2019**, *28*, 033005. [CrossRef]
31. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576.
32. Johnson, J.; Alahi, A.; Li, F. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 694–711.
33. Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1501–1510.
34. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.