Research Article

High-Capacity Image Steganography Algorithm Based on Image Style Transfer

Xinliang Bi ^{1,2}, Xiaoyuan Yang ^{1,2}, Chao Wang ¹ and Jia Liu ^{1,2}

¹Engineering University of PAP, College of Cryptographic Engineering, Xi'an 710086, China

²Network and Information Security Key Laboratory of PAP, Xi'an 710086, China

Correspondence should be addressed to Xiaoyuan Yang; wjcdxxy@163.com

Received 6 July 2021; Accepted 13 September 2021; Published 28 September 2021

Academic Editor: Zhili Zhou

Copyright © 2021 Xinliang Bi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Steganography is a technique for publicly transmitting secret information through a cover. Most of the existing steganography algorithms are based on modifying the cover image, generating a stego image that is very similar to the cover image but has different pixel values, or establishing a mapping relationship between the stego image and the secret message. Attackers will discover the existence of secret communications from these modifications or differences. In order to solve this problem, we propose a steganography algorithm ISTNet based on image style transfer, which can convert a cover image into another stego image with a completely different style. We have improved the decoder so that the secret image features can be fused with style features in a variety of sizes to improve the accuracy of secret image extraction. The algorithm has the functions of image steganography and image style transfer at the same time, and the images it generates are both stego images and stylized images. Attackers will pay more attention to the style transfer side of the algorithm, but it is difficult to find the steganography side. Experiments show that our algorithm effectively increases the steganography capacity from 0.06 bpp to 8 bpp, and the generated stylized images are not significantly different from the stylized images on the Internet.

1. Introduction

While the advancement of social informatization has brought tremendous changes to people's lifestyles, it has also brought some hidden dangers. For example, more personal information is exposed on the Internet, and the privacy of information transmission is challenged. As one of the important technologies to protect transmission security, steganography has received widespread attention. Unlike cryptography, steganography pays more attention to protecting the invisibility of secret communication, while cryptography pays more attention to protecting the invisibility of secret information.

Image steganography is a steganography based on images. Image steganography can be roughly divided into two categories: one is the traditional image steganography technology, and the other is the image steganography technology based on deep learning. Traditional image steganography algorithms embed secret information by

changing the pixel values of the image and ensure that the embedding has a minimal impact on the cover image by minimizing the loss function [1–3]. The biggest difference between these algorithms is the design of the loss function. The difference in the loss function can have a huge impact on the steganography effect. Traditional image steganography algorithms require manual design of the steganography strategy and require sufficient expertise of the designer. As the image is modified, it will inevitably leave modification traces on the image and cause some statistical features of the image to change [4–6], increasing the possibility of secret communication exposure. With the development of deep learning techniques, people started to use deep neural networks to minimize the loss between cover image and stego image and use a large amount of data to automate the process of finding a suitable steganography strategy. Image steganography based on deep learning attempts to find a suitable location for hiding information in an image through a neural network and evaluate the amount of information

that can be hidden [7] or directly generate an image through secret information [8]. Compared with traditional image steganography technology, image steganography technology based on deep learning has the characteristics of fast design and strong performance improvement. For example, through deep learning, the color image is hidden in another color image, which greatly increases the steganography capacity [9]; by adding a noise layer, the robustness of secret information extraction is enhanced [10].

Due to the popularity of cameras, mobile phones, and other types of shooting equipment and the improvement of network transmission capabilities, it has become more convenient for people to obtain and transmit images. Sometimes, people prefer to transmit readily available image information instead of text information, and these pieces of image information can also be retained as evidence. It also promotes the production and development of high-capacity image steganography technology. High-capacity image steganography is a technology that hides one or more color or grayscale images in a color or grayscale image. Although high-capacity image steganography technology can greatly increase the steganography capacity of the algorithm, it also has certain problems. For example, if an attacker can obtain a cover image and a stego image at the same time, he can obtain the residual image from the two images. Part of the semantic information of the secret image will be discovered. In response to this problem, some scholars have proposed that the secret image can be encrypted before embedding [11, 12], although this method can solve the problem of the semantic information exposure of the secret image. However, if an attacker can obtain cover and stego images, he may think that the stego image is the result of cover image modification and will still question the existence of secret communication.

Image processing is a technique that is often used by us, such as image compression and image denoising. Image processing will cause changes in the pixel values or statistical characteristics of the image. We can disguise the image changes caused by steganography as such changes, making image steganography and image processing indistinguishable. Some scholars have proposed a neural network STNet [13] that can embed secret information in the process of image style transfer. This algorithm combines the process of steganography with the process of style transfer. The algorithm more shows the function of style transfer and hides the steganography function. At the same time, the generated stylized images are indistinguishable from the stylized images on the Internet. Even if the attacker can obtain the cover image and the stego image, the secret information cannot be extracted from the residual images of the two. This greatly improves the security of the steganography algorithm. On this basis, we proposed ISTNet. We improve the decoder of STNet, fuse the secret image features with the adaptive instance normalization (AdaIN) [14] layer results in multiple scales, and hide a grayscale image during the image style transfer process. On the basis of maintaining the security of STNet, the steganography ability is improved. Figure 1 is the difference between ISTNet and STNet. On the left is ISTNet, and on the right is STNet. STNet receives content images,

style images, and secret information, while ISTNet receives content images, style images, and secret images.

The following are our three main contributions:

- (1) We propose a new steganography network that can hide a grayscale image into another color image of the same size during image style transfer. Experiments have proved that it is difficult to find the difference between our stego image and the image generated by the other image style transfer network.
- (2) Compared with STNet, the algorithm that we proposed greatly improves the steganography capacity. In the STNet, the steganography capacity is only 0.06 bits per pixel (bpp), and ISTNet can hide a grayscale image of the same size.
- (3) In this paper, the secret information embedding process and the image translation process are integrated, and the attacker cannot distinguish whether the image processing process is image steganography or image translation.

The rest of this paper is organized as follows. In Section 2, we introduced some steganography algorithms based on deep learning. In Section 3, we introduced our proposed algorithm. Section 4 presents the experimental results to verify our proposed algorithm. Finally, the conclusion is drawn in Section 5.

2. Related Work

In this section, we first introduce some deep learning steganography algorithms based on modification, then introduce some deep learning steganography algorithms that directly generate stego images without modification, and finally introduce some high-capacity image steganography algorithms.

2.1. Image Steganography Algorithm Based on Modification.

In 2014, after Generative Adversarial Networks (GAN) [15] was proposed, some scholars quickly introduced it into the field of image steganography. Hayes and Danezis [16] designed a simple GAN structure, which includes a generator, an extractor, and a steganalyzer. The generator receives the cover image and secret information and generates a stego image. The extractor extracts the secret message from the stego image. The steganalyzer receives the stego image and the cover image and learns to distinguish between the two. Through adversarial training, a powerful generator and steganalyzer are produced. Although the steganography algorithm performance of the generator is slightly worse than HUGO, S-UNIWARD, and WOW, the steganalyzer is not as good as ATS [17]. But its algorithm design is simple, showing the prospect of GAN in the field of image steganography and its huge potential. Unlike the algorithm of Hayes and Danezis, Volkonskiy et al. [18] proposed a more complex GAN structure. They no longer use GAN as an embedder but use GAN to generate cover images and use ± 1 -embedding [19] algorithm to embed secret information in cover images. They use the discriminator to improve the

FIGURE 1: The difference between STNet and ISTNet. Compared with STNet, ISTNet has a larger steganography capacity. It can hide a grayscale image while completing the image style transfer, while STNet can only hide brief secret information.

quality of image generation and use the steganalyzer to improve the algorithm's ability to resist steganalysis. Although the steganalyzer of this algorithm has been greatly improved, the generated image is not the same as the normal image, the generated quality is poor, and it is easy to be found to be the generated image at a glance. Wang et al. [20] proposed a network structure similar to Hayes et al., but because the design of the network is more complex, the performance of the generated images against steganalysis even exceeds HUGO, WOW, and S-UNIWARD. In 2019, Volkhonskiy et al. improved its algorithm and added a secret key [21]. The extractor needs a secret key to correctly extract the secret information from the stego image, which improves the security of the algorithm. Tang et al. [7] and Yang et al. [22] have different ideas from the above two. They used GAN to analyze where the information can be hidden in the image and how many secrets can be hidden and used the more advanced steganalyzer Xu-net [23] for adversarial training. At this point, the performance of this algorithm has surpassed S-UNIWARD.

In the above algorithm, the cover image will be modified or the cover image will be required as input. This type of steganography algorithm is more like an improvement to the traditional algorithm, using deep learning technology to learn effects that are difficult to achieve by manual design or using deep learning technology to find the optimal solution. However, this algorithm will still leave traces of modification in the image, and the security needs to be improved.

2.2. Image Steganography Algorithm Based on No Modification. As DCGAN [24] and other neural networks that generate images driven by noise are proposed, Hu et al. [8] proposed to map secret information to noise and use the noise to generate stego images. At the same time, a separate extractor is trained to extract secret information from stego images. Compared with the image steganography algorithm based on modification, the advantage of this algorithm is that there is no cover image. The generated image is an unmodified stego image, and the algorithm has high security. Zhang et al. [25] also designed an algorithm to map the secret information to a stego image. Unlike the mapping

method of Hu et al., Zhang et al. map the secret information to an image label and then use the label to generate a stego image. Compared with extracting secret information, tag extraction has characteristics such as robustness and stability, which makes the extraction accuracy of Zhang et al.'s algorithm higher than that of Hu et al., but its steganography capacity is much smaller than that of Hu et al. The above two algorithms both directly generate stego images through secret information, which solves the problem of long search time for stego images in the traditional unmodified image steganography algorithm, but there is still a gap between the generated images and the real images. Meng et al. [26] proposed an unmodified algorithm based on Faster R-CNN [27]. They use Faster R-CNN to identify targets in the image and give each target a different code. According to the order size of the detection frame, the codes represented by the targets are rearranged to form complete secret information. Since the Faster R-CNN algorithm has high robustness in target recognition, the algorithm is also robust and can resist a variety of attacks. Liu et al. [28] use feature extractor of DenseNet [29] to map the image into features and then transform the image into secret information.

The above four algorithms directly map the secret information to the generated image or use the feature extraction algorithm to extract the secret information from the stego image. This type of algorithm has high security because the transmission is completely an unmodified normal image. But it also has problems such as slow searching for secret images, unrealistic generated images, and low capacity.

2.3. High-Capacity Image Steganography Algorithms. With the development of deep learning and the improvement of computing resources, some scholars try to hide a grayscale image or a color image in a color image. Baluja pays more attention to improving the steganography ability of images [9]. Through neural networks, the steganography capacity of the algorithm is greatly improved. A color image can be hidden in another color image of the same size, and the steganography capacity is increased to 24 bpp. ur Rehman et al. [30] use a different neural network to increase the steganography capacity to 8bpp. Compared with the

traditional image steganography scheme, the schemes proposed by Zhang et al. [31] and Duan et al. [32] greatly increase the steganography capacity. However, this type of steganography scheme also has certain problems. For example, the semantic information of the secret information can be obtained from the residual image of the cover image and the stego image. Figure 2 is the cover image and stego image generated by the algorithm proposed by Baluja. The residual image is the difference between the cover image and the stego image. From the residual image, we can clearly find some information from the secret image. In order to solve this problem, Duan et al. [12] and Sharma et al. [11] proposed to encrypt the secret image first and then embed the encrypted secret image into the cover image. Although these methods can hide the specific content of the secret image, the attacker may still think that the stego image is modified from cover image.

In response to these problems, STNet embeds secret information in the process of image translation. Compared with the above algorithms, the cover image and stego image of this algorithm are completely two types of images. Even if the attacker obtains these two images, he still cannot obtain the secret information. At the same time, STNet can also disguise the embedding process of secret information as an image style transfer process instead of image steganography, which reduces the probability of secret communication being discovered and increases the overall security of the scheme. We also hide the secret information in the process of image translation. Unlike the STNet, we hide a grayscale image of the same size as the cover image during the image style transfer process instead of text information.

3. Proposed Algorithm

The overall structure of the ISTNet is shown in Figure 3, which consists of two parts, the steganography network and the extraction network. The steganography network contains an encoder consisting of VGG network, a feature extraction network, and a decoder. The steganography network combines the features of the style image to convert the content image into an image with the style image features and embeds the secret image in this process. In our algorithm, the cover image is also the content image. The extraction network extracts the secret image from the stego image.

3.1. Steganography Network. The encoder consists of two trained VGG networks, whose structure is shown in Figure 4. The VGG network is the first twenty-one layers of the VGG-19 network [33] (including the convolutional layer, the ReLU layer, and the max pooling layer). The parameters of the VGG network are pretrained in the ImageNet dataset. Since these pretrained VGG networks have been trained on a large number of images, their feature extraction effects are better. Using a pretrained feature extraction network can reduce training expenses and training time, and the difficulty of convergence is also reduced. We use the trained VGG

network to extract high-dimensional features from content images and style images. The two features will be coded into one feature through the AdaIN layer. The decoder then maps the features back to the image space. At this time, the secret image has been embedded in the cover image. The calculation process of the AdaIN layer is shown in equation (1). In the equation, x represents the content image, y represents the style image, $\mu(\bullet)$ represents the mean value of image pixels, and $\sigma(\bullet)$ represents the variance of image pixels. Compared with the convolutional layer, AdaIN has a better stylization effect and does not require training parameters, which makes the overall training of the network faster and takes less time.

$$\text{AdaIN} = \sigma(y) \frac{x - \mu(x)}{\sigma(x)} + \mu(y). \quad (1)$$

STNet directly embeds the secret information into the results produced by the AdaIN layer. Although its steganography process will not cause great loss to the effect of image style transfer, its steganography capacity is too low and only 0.06 bpp. We have expanded its steganography capacity to hide a grayscale image instead of short secret information. We refer to the network structure of U-net [34], design a feature extractor to extract features from the secret image, and fuse the extracted features with the process features of the decoder at the same size. Under the circumstance that the effect of image style transfer remains unchanged, the steganography capacity is greatly increased.

The details of the decoder are shown in Figure 5. In the figure, the solid-line box represents the operation taken on the feature, and the dashed-line box represents the size of the current feature. Unlike the decoder of STNet, we make full use of the feature of the secret image in the decoding process to ensure higher extraction accuracy. In this paper, a new feature extraction network is designed to extract the features of secret images. The right side of Figure 5 is the structure of the feature extraction network of the secret image. It contains three convolutional layers, and each convolutional layer adds a ReLU layer. Each convolution process will reduce the size of the image by half, and the features extracted by each convolution layer will be combined with features of the same size during the decoding process. The left side of Figure 5 is the feature decoding network, where the convolutional layer does not change the image size. The feature size is expanded by interpolation expansion, and each interpolation expansion operation will double the feature size.

3.2. Extraction Network. When the feature size is reduced, it will inevitably cause the loss of stego image information. Therefore, we did not reduce the feature size, and we hope that the feature can always maintain the original size to ensure the full use of the stego feature. Moreover, what we need to extract is the secret image instead of the secret information, so it is not very necessary to gradually reduce the size of the feature to generate the secret information. The extraction network structure is shown in Figure 6. The stego

FIGURE 2: Residual image of cover image and stego image. The following three images are the residual images of the blue channel, the green channel, and the red channel. Whether it is the residual image or the residual image of each channel, the outline of the secret image can be found.

FIGURE 3: Overall network structure. ISTNet consists of a steganography network and an extraction network. The steganography network receives cover images and style images and generates a stylized stego image, and the extraction network receives the stego image and extracts the secret image from it.

image will be processed by six convolutional layers and ReLU functions. After each convolutional layer, there is a batch normalization (BN) layer to improve the network convergence speed. The convolution kernel parameters of each convolution layer are set to 3×3 , and the step size parameter is set to 1, so in the process of extracting the secret image, the size of the feature has not changed, but the number of channels is constantly changing.

3.3. Loss Function. The loss in this paper mainly includes two parts: one is the loss function L_{Se} of the steganography network, and the other is the loss function L_E of the extraction network. In the process of image embedding, the steganography network should not only minimize the impact of image embedding on image style transfer but also enable the extraction network to extract the secret image from the stego image as accurately as possible. Therefore,

FIGURE 4: The details of the encoder. Encoder extracts the features of cover image and style image by two identical VGG networks.

FIGURE 5: The details of the decoder. During the decoding process, the decoder will continuously merge the features of the secret image with the stylized features.

FIGURE 6: The details of the extraction network. The extraction network is composed of multiple convolutional layers, and the features only change in dimension and do not change in size during the extraction process.

when designing the loss function L_{Se} , both the steganography effect and the extraction effect are considered. The loss function L_{Se} is

$$L_{Se} = L_c + \alpha L_s + \beta L_e. \quad (2)$$

In the above equation, L_c represents the content loss, L_s represents the style loss, and L_e represents the extraction loss. α and β are the weights of style loss and extraction loss, respectively, which are used to balance L_c , L_s , and L_e . We use the Euclidean distance to measure the content loss of the image:

$$L_c = \|f(g(t)) - t\|_2. \quad (3)$$

In equation (3), t represents the result of AdaIN layer, $f(\bullet)$ represents the VGG network, and $g(\bullet)$ represents the decoder in the steganography network. Through this loss function, the feature of the generated image is constrained to be as consistent as possible with the feature t processed by AdaIN to improve the content similarity between the stego image and the content image. L_s is given as

$$L_s = \sum_{i=1}^L \|\mu(\phi_i(g(t))) - \mu(\phi_i(s))\|_2 + \sum_{i=1}^L \|\sigma(\phi_i(g(t))) - \sigma(\phi_i(s))\|_2. \quad (4)$$

In equation (4), s represents the style image, $\mu(\bullet)$ represents the mean value, and $\sigma(\bullet)$ represents the variance. L represents the set of specific layers in the neural network. $\phi_i(\bullet)$ represents the output of the i -th layer of the VGG network. i we used in this paper are $relu_{1_1}$, $relu_{2_1}$, $relu_{3_1}$, and $relu_{4_1}$, which are the 2nd, 7th, 12th, and 21st layers of the VGG network.

$$L_e = \frac{1}{W \times H} \sum_{m=1}^{W \times H} (I_{sr_m} - I_{er_m}). \quad (5)$$

We use mean square error (MSE) to evaluate the loss between the secret image and the extract image. MSE is shown in equation (5), where W and H represent the width and height of the secret image, I_{sr} represents the secret image, and I_{er} represents the extracted image.

The extractor network only needs to extract the secret image as accurately as possible from the stego image. Its loss function L_E is the same as L_e :

$$L_E = \frac{1}{W \times H} \sum_{m=1}^{W \times H} (I_{sr_m} - I_{er_m}). \quad (6)$$

4. Experiments

In this section, we use a lot of experiments to verify that the images generated by ISTNet are indistinguishable from other stylized images on the Internet and also analyze the influence of different secret images on stego images. The difference between the stego image generated by ISTNet and some other high-capacity steganography algorithms is analyzed, and the steganography capacity is compared with other steganography algorithms based on deep learning. We use the current advanced deep learning-based steganalysis algorithm SRNet [35] and the steganalysis tool StegExpose [36] to analyze the difference between our algorithm and other style transfer algorithms, proving that the result of ISTNet is not much different from other style transfer algorithms. Finally, we use peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) to evaluate the extraction loss.

4.1. Experiments Settings. We experimented with our proposed algorithm on a computer with Windows 10 operating system. The computer is equipped with an RTX2080Ti graphics card with 11 GB of video memory, and the computing framework is CUDA10.0. The deep learning framework we use is Pytorch 1.5.0, and the programming language is Python 3.6.8.

The dataset we use is ukiyoe2photo [37], which contains 825 Ukiyo-e images and 7563 natural landscape images. The training dataset includes 562 Ukiyo-e images and 6287 natural landscape images, and the test dataset includes 263 Ukiyo-e images and 1276 natural landscape images. The secret images are 6287 images randomly selected from the VOC2012 dataset. In order to facilitate training, the size of these images is adjusted to 256×256 , and the secret images

are adjusted to grayscale images. The parameter α is set to 10, the parameter β is set to 1, the learning rate is set to 0.001, and the epochs are set to 100.

4.2. Stylized Comparison. Figure 7 is a comparison between the results of our steganography algorithm and the results of other style transfer algorithms. We chose three different pairs of content images and style images to observe the comparison of our generated images with those generated images by other algorithms. The two algorithms we compared both use part of the feature extraction in the VGG-19 network as the encoder, which is very similar to our algorithm. From the results, our algorithm is somewhat different from the results of Huang and Belongie [14] and Li et al. [38], but they both complete the image translation task. Although the transfer results of our algorithm are different from other algorithms, the style transfer images produced by other style transfer algorithms on the Internet are also different. It is difficult to distinguish our algorithm from other algorithms on the Internet.

Although our algorithm can embed a grayscale image while completing the style transfer task, our decoder will be combined with the features of the secret image many times during the feature decoding process. The change of the secret image may have a serious impact on the decoding. If the secret image can seriously affect the stego image, it will produce a mapping between the secret image and the stego image, and the attacker can derive the secret image from the stego image, which will pose a great threat to the security of the algorithm. In order to explore the influence of the secret image change on the stego image, we used the same content image and style image pairs and different secret images for experiments. The results are shown in Figure 8.

It can be found from Figure 8 that visually different secret images will not have a strong effect on the stego image, but the visual evaluation is more subjective, so we use PSNR and SSIM to analyze the difference between stego images generated by different secret images. The analysis results are shown in Figure 9, the x -axis is the image number, and the y -axis is the PSNR value and SSIM value of the different stego images compared with the first stego image. We used a total of 100 stego images generated from different secret images for experiments.

From the subjective feeling, the PSNR value, and the SSIM value, we can find that the change of the secret image will not have a serious impact on the stego image, and the attacker cannot directly infer the secret image information from the stego image.

4.3. Comparison with High-Capacity Steganography Algorithm. Among the existing algorithms, there are already some algorithms that can hide another color image or grayscale image in a color image. However, these algorithms have some shortcomings; that is, their cover image and stego image have a high degree of similarity. It is difficult for the human eye to distinguish the difference between the two. They are also very similar in statistical characteristics, but the specific pixel values of the image are different. If the attacker

can obtain the cover image and the stego image at the same time, he can find that the stego image is modified from the cover image, which undoubtedly poses a threat to the security of communication. Compared with these algorithms, our proposed algorithm will convert the cover image to another style while embedding the secret image. Even if the attacker obtains the cover image and the stego image at the same time, it is likely that the sender has just changed the style of the image instead of embedding the secret image. Compared with other embedding algorithms, we rationalize the modification of the image caused by embedding and disguise the modification as a result of style transfer.

Figure 10 is a comparison between our proposed algorithm and the algorithms proposed by ur Rehman et al. [30], Zhang et al. [31], and Baluja [9]. It can be seen from the figure that the algorithm proposed by Rehman not only modifies the cover image but also causes the color of the image to be distorted. Although the algorithm of Zhang et al. and Baluja did not cause color distortion, it also caused a certain amount of image loss. As mentioned in “related work,” this will cause the attacker to discover part of the secret image information from the residual between the cover image and the stego image. In comparison, our proposed algorithm has completed the task of style transfer while embedding secret images and can embed secret images in disguise as style transfer. Even if the attacker can obtain the stego image and the cover image at the same time, he cannot obtain the secret image from the residual image.

4.4. Capacity. We use bpp to evaluate steganography capacity. The algorithm we proposed can hide a grayscale image in a color image with a size of 256×256 . On average, 8 bits of secret information can be hidden per pixel, and the steganography capacity is 8 bpp. Compared with the steganography algorithms where the embedded secret information is also an image, we can achieve the same steganography capacity as Rehman et al. and Zhang et al. Compared with Baluja, the steganography capacity is reduced, but our algorithm also completes the style transfer task, so a certain steganography capacity will be sacrificed. The algorithm of Hidden StegaStamp, STNet, and SteganoGAN is to hide secret information instead of an image. Compared with their algorithm, our steganography capacity has a certain improvement. We have improved STLNet. Compared with it, we hide a gray image instead of short secret information. On the basis of ensuring that its other characteristics remain unchanged, the steganography capacity has been greatly improved. The comparison of steganography capacity is shown in Table 1, where W and H represent the width and height of the cover image.

4.5. Security. There are many style transfer images on the Internet, and the images generated by our algorithm look no different from the style transfer images on the Internet. In order to better verify that our style transfer images are not much different from the style transfer images on the Internet, we used the steganalysis network SRNet and the steganalysis tool StegExpose to analyze the difference

FIGURE 7: Comparison of the stylized image of ISTNet with the results of other style transfer networks. Although our algorithm needs to embed an image at the same time as the style transfer, it does not have a great impact on the style transfer.

FIGURE 8: The influence of different secret images on stego images. The change of the secret image did not have a major impact on the stego image.

FIGURE 9: Comparison of PSNR and SSIM values between different stego images. The PSNR and SSIM values between different stego images are similar, which proves that the change of secret does not have a major impact on the content and style of the stego image.

between the stego image generated by ISTNet and other style transfer images. We mainly compared our algorithm with the algorithms of Huang et al., Li et al., Zhu et al., and

Sanakoyeu et al. [42]. We used 100 content images and corresponding stylized images for experiments, and the style images are all the Ukiyo-e images. We used SRNet and

FIGURE 10: Results generated by different high-capacity steganography algorithms. Compared with other algorithms where the embedded secret information is an image, we have broken the convention. Our cover image and stego are no longer very similar but very different. ISTNet is more likely to be considered as a style transfer algorithm rather than a steganography algorithm.

TABLE 1: Steganography capacity comparison.

Algorithm	Secret information size	Cover image size	Capacity (bpp)
WOW [2]	6554 bit	128×128	0.400
S-UNIWARD [1]	6554 bit	128×128	0.400
Hidden [10]	52 bit	16×16	0.203
StegaStamp [39]	100 bit	400×400	6.25×10^{-4}
STNet [13]	Arbitrary length	Arbitrary size	0.06
SteganoGAN [40]	$W \times H \times 6$ bit	$W \times H \times 3$	6
Qin et al. [41]	$W \times H \times 6$ bit	$W \times H \times 3$	6
Rehman et al. [30]	$256 \times 256 \times 1$	$256 \times 256 \times 3$	8
Zhang et al. [31]	$256 \times 256 \times 1$	$256 \times 256 \times 3$	8
Baluja [9]	$256 \times 256 \times 3$	$256 \times 256 \times 3$	24
Ours	$256 \times 256 \times 1$	$256 \times 256 \times 3$	8

StegExpose to analyze these 200 images. The obtained SRNet and StegExpose analysis results of stylized images generated by different algorithms are shown in Figures 11 and 12.

It can be observed from the figure that SRNet has some differences in the judgment of stylized images generated by different algorithms, but the difference is not very large, and it is difficult to distinguish the difference between our stego images and other stylized images. In the receiver operating characteristic (ROC) curve drawn by the StegExpose judgment result, it can also be seen that the ROC curves of different stylized images are near the diagonal, and the ROC curve of our algorithm is slightly different from other stylized images. At the same time, there are more types of

stylized images on the Internet than in comparison experiments. It is very difficult to distinguish our stylized images from so many stylized images on the Internet.

4.6. Loss of Extracted Image. The extraction accuracy of secret information is also one of the important bases for evaluating steganography algorithms. In order to compare the accuracy of our algorithm for extracting secret images, we used PSNR and SSIM to evaluate the difference between the extract image and the secret image and compared them with several steganography algorithms based on deep learning. We used 100 pairs of secret images and extract

FIGURE 11: The result of the SRNet. From the results of the steganalyzer, our proposed algorithm is not different from the style transfer algorithm.

FIGURE 12: The result of the steganalyzer StegExpose. From the results of the steganalyzer, we find that the ROC curve of our algorithm is very similar to the ROC curve of other algorithms and can make the steganalyzer more classification errors.

TABLE 2: Secret image loss comparison.

Algorithm	PSNR	SSIM
Duan et al. [32]	40.66	0.98
Rehman et al. [30]	36.59	0.96
Zhang et al. [31]	36.58	0.95
Duan et al. [12]	40.57	0.96
Duan et al. [43]	37.09	0.98
Ours	30.48	0.75

FIGURE 13: Comparison of secret image and extract image. Compared with secret images, extract images have a certain loss, but it does not affect the understanding of its image semantics and other pieces of information.

images to calculate PSNR and SSIM and take the average as the final result. The comparison results are shown in Table 2.

Experiments have proved that, compared with other algorithms, the extraction effect of our algorithm is not very good. However, transferring images is different from transferring secret information. A high accuracy rate is required for the transmission of secret information because the error or lack of keywords of the secret information will have a serious impact on the understanding of the secret information. The loss when extracting the image, such as noise or changes in the pixel value of some pixels, will not affect the overall image understanding. Compared with other algorithms, our algorithm has to complete the image translation task while embedding the secret image, which will inevitably have an impact on the extraction of the secret image.

We have listed several secret image and extract image pairs as shown in Figure 13. It can be seen from the figure that although the PSNR and SSIM of our algorithm are relatively low, the difference between the secret image and the extract image is not big, and the impact on the image understanding is minimal.

5. Conclusion

We make improvements on the basis of STNet so that the features of the secret image are merged with the results generated by AdaIN in the decoding process in multiscale, and an image is hidden instead of short secret information in the image style transfer. The steganography capacity has been increased from 0.06 bpp to 8 bpp. Experiments have proved that our stylized images are not very different from the stylized images on the Internet. The attacker cannot find the traces of modification from the stego image, nor can he find the secret image information from the residual image of the stego image and the cover image.

Data Availability

Some or all data, models, or code generated or used during the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61872384).

References

- [1] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, 13 pages, 2014.
- [2] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proceedings of the 2012 IEEE International workshop on information forensics and security (WIFS)*, December 2012.
- [3] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proceedings of the International Workshop on Information Hiding*, June 2010.
- [4] T. Bhuiyan, A. H. Sarower, R. Md Karim, and M. Md Hassan, "An image steganography algorithm using LSB replacement through XOR substitution," in *Proceedings of the 2019 International Conference on Information and Communications Technology (ICOLACT)*, July 2019.
- [5] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2021.

- [6] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 685–696, 2020.
- [7] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.
- [8] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38303–38314, 2018.
- [9] S. Baluja, "Hiding images in plain sight: deep steganography," *Advances in Neural Information Processing Systems*, vol. 30, pp. 2069–2079, 2017.
- [10] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 657–672, Munich, Germany, September 2018.
- [11] K. Sharma, A. Aggarwal, T. Singhanian, and G. Deepak, "Hiding data in images using cryptography and deep neural network," 2019, <https://arxiv.org/abs/1912.10413>.
- [12] X. Duan, D. Guo, N. Liu, B. Li, M. Gou, and C. Qin, "A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network," *IEEE Access*, vol. 8, pp. 25777–25788, 2020.
- [13] Z. Wang, N. Gao, X. Wang, J. Xiang, and G. Liu, "STNet: a style transformation network for deep image steganography," *Neural Information Processing*, Springer, in *Proceedings of the International Conference on Neural Information Processing*, pp. 3–14, December 2019.
- [14] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, Venice, Italy, October 2017.
- [15] I. J. Goodfellow, J. Pouget-Abadie, and M. Mirza, "Generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.
- [16] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," 2017, <https://arxiv.org/abs/1703.00371>.
- [17] D. Lerch-Hostalot and D. Megías, "Unsupervised steganalysis based on artificial training sets," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 45–59, 2016.
- [18] D. Volkhonskiy, B. Borisenko, and E. Burnaev, "Generative adversarial networks for image steganography," 2016.
- [19] A. D. Ker, "Resampling and the detection of LSB matching in color bitmaps," in *Proceedings of the Security, steganography, and watermarking of multimedia contents VII*, vol. 5681, pp. 1–15, San Jose, California, USA, January 2005.
- [20] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "SStGAN: self-learning steganography based on generative adversarial networks," in *Proceedings of the International Conference on Neural Information Processing*, pp. 253–264, Springer, Berlin, Cham, December 2018.
- [21] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," in *Proceedings of the Twelfth International Conference on Machine Vision (ICMV 2019)*, vol. 11433, p. 11433M, Amsterdam, Netherlands, November 2020.
- [22] J. Yang, K. Liu, X. Kang, and E. Wong, "Spatial image steganography based on generative adversarial network," 2018, <https://arxiv.org/abs/1804.07939>.
- [23] Y. Qian, J. Dong, W. Wang, and T. Tan, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proceedings of the 2016 IEEE international conference on image processing (ICIP)*, pp. 2752–2756, IEEE, Phoenix, Arizona, USA, September 2016.
- [24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, <https://arxiv.org/abs/1511.06434>.
- [25] Z. Zhang, G. Fu, R. Ni, J. Liu, and X. Yang, "A generative method for steganography by cover synthesis with auxiliary semantics," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 516–527, 2020.
- [26] R. Meng, S. G. Rice, J. Wang, and X. Sun, "A fusion steganographic algorithm based on faster R-CNN," *Computers, Materials & Continua*, vol. 55, no. 1, pp. 1–16, 2018.
- [27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
- [28] Q. Liu, X. Xiang, J. Qin, and Y. Tan, "Coverless image steganography based on DenseNet feature mapping," *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–18, 2020.
- [29] G. Huang, Z. Liu, L. Van Der Maaten, and W. Kilian, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.
- [30] A. ur Rehman, R. Rahim, S. Nadeem, and S. ul Hussain, *End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography*, European Conference on Computer Vision, Springer, Cham, Switzerland, pp. 723–729, 2018.
- [31] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8559–8575, 2019.
- [32] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a U-Net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.
- [33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
- [34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, Munich, Germany, October 2015, Lecture Notes in Computer Science.
- [35] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.
- [36] B. Boehm, "Stegexpose-A tool for detecting LSB steganography," 2014, <https://arxiv.org/abs/1410.6656>.
- [37] J. Y. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, Venice, Italy, October 2017.
- [38] Y. Li, C. Fang, J. Yang, Z. Wang, L. Xin, and M. Yang, "Universal style transfer via feature transforms," 2017, <https://arxiv.org/abs/1705.08086>.
- [39] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2117–2126, Seattle, WA, USA, June 2020.

- [40] K. A. Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "SteganoGAN: high capacity image steganography with GANs," 2019, <https://arxiv.org/abs/1901.03892>.
- [41] J. Qin, J. Wang, Y. Tan, H. Huang, X. Xiang, and Z. He, "Coverless image steganography based on generative adversarial network," *Mathematics*, vol. 8, no. 9, p. 1394, 2020.
- [42] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time hd style transfer," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 698–714, Munich, Germany, October 2018.
- [43] X. Duan, L. Nao, G. Mengxiao et al., "High-capacity image steganography based on improved FC-DenseNet," *IEEE Access*, vol. 8, pp. 170174–170182, 2020.