mathematics

MDPI

Article

Coverless Image Steganography Based on Generative Adversarial Network

Jiaohua Qin [1,2,*], Jing Wang [2], Yun Tan [2], Huajun Huang [1], Xuyu Xiang [2] and Zhibin He [2]

[1] College of Information Technology and Management, Hunan University of Finance and Economics, Changsha 410205, China; hhj0906@163.com

[2] College of Computer Science and Information Technology, Central South University of Forestry and Technology, Changsha 410004, China; wangjinginSZU@gmail.com (J.W.); tantanyun@hotmail.com (Y.T.); xyuxiang@163.com (X.X.); hzb919086716@163.com (Z.H.)

* Correspondence: qinjiaohua@csuft.edu.cn

Received: 3 May 2020; Accepted: 9 July 2020; Published: 20 August 2020

Abstract: Traditional image steganography needs to modify or be embedded into the cover image for transmitting secret messages. However, the distortion of the cover image can be easily detected by steganalysis tools which lead the leakage of the secret message. So coverless steganography has become a topic of research in recent years, which has the advantage of hiding secret messages without modification. But current coverless steganography still has problems such as low capacity and poor quality .To solve these problems, we use a generative adversarial network (GAN), an effective deep learning framework, to encode secret messages into the cover image and optimize the quality of the steganographic image by adversaring. Experiments show that our model not only achieves a payload of 2.36 bits per pixel, but also successfully escapes the detection of steganalysis tools.

Keywords: coverless steganography; deep learning; generative adversarial network

1. Introduction

Since the invention of the Internet, technology has developed rapidly. The emergence of multimedia information such as images, audio and video has brought convenience to society [1] but it has also resulted in the illegal wiretapping, interception, tampering or destruction of important and sensitive information related to politics, military, finance and business, bringing huge losses to society. Therefore, information hiding technology has emerged [2,3]. With the development of this technology, the corresponding steganographic detection technology has also evolved. The traditional approaches, which adopt artifacts, tend to be easily detected by automated steganalysis tools and, in extreme cases, by human eyes, which poses the challenge of information hiding.

To solve this problem, researchers proposed a new information hiding method—coverless steganography—in 2015. Compared with the traditional approaches, which need to adopt the specified cover image for embedding the secret data, such as Highly Undetectable SteGO (HUGO) and JPEG compression [4–7], the coverless steganography no longer modifies the cover images, which is why it is called coverless. It is achieved by means of mapping with secret information. Even if the image is intercepted, it is hard to detect the presence of a message. Therefore, coverless steganography can naturally resist steganalysis tools. At present, existing coverless steganography is divided into two categories according to the steganographic principle—mapping-based [8,9] and synthetic-based methods [10]. The coverless image steganography based on mapping rules was first proposed by Zhou [11]. Each image represented an 8-bit pixel and was divided into nine blocks, and the feature sequence was calculated from the relationship between the mean values of adjacent block pixels. Zheng et al. [12] proposed an image steganography algorithm based on invariant features

(SIFT). Unlike Zhou, Zheng used feature sequences generated by SIFT features, which enhanced the robustness of the system. Recently, Zhou et al. [13] proposed a method based on SIFT and Bag-of-Features (BOF). Compared with Reference [11], this method can better resist rotation, zoom, brightness change and other attacks, but the ability to resist translation, filter and shear is still limited.

The instance-based texture synthesis algorithm is a hotspot of current texture synthesis algorithms, which synthesizes new texture images by resampling the original images. The new texture image can be of any size, and its local appearance is similar to the original image. Otori [14,15] and others pioneered a steganographic algorithm based on pixel-based texture synthesis. First, they encoded the secret information into a colored dot pattern and then automatically draw a pattern from the sample image on the coat texture image to mask its existence and natural texture mode. Wu et al. [16] proposed an image steganography algorithm based on patch texture synthesis. Firstly, an overlap area will be generated during the synthesis process, and the mean square error of the overlap area and the candidate block will be calculated so as to sort the candidate blocks. Finally, the candidate blocks identical to the secret information sequence number are synthesized into the overlapping area to hide the secret information. However, if the method needs to hide more information, the hidden ability will drop. Inspired by the the marble deformation texture synthesis algorithm, Xu et al. [17] proposed a reconfigurable image steganography algorithm based on texture deformation. In order to hide the secret information, the secret image is reversibly twisted to synthesize different marble textures, but the robustness of the algorithm is limited.

Coverless information hiding is still a relatively new field. Compared with other information hiding technologies, its theoretical research and technical maturity still have some gaps, and there are still some problems such as low hiding capacity and efficiency. With the advent of deep learning [18–20], a new method of image steganography approaches is emerging [21–24]. The first set of deep learning approaches to steganography were from Baluja [22]. They used neural networks to combine a cover image and a secret message into a steganographic images but their images showed a strong spatial correlations, and convolutional neural network (CNN) training will use this feature to hide images in the images. So, the model trained in this way cannot be applied to arbitrary data. The emergence of generative adversarial networks (GANs)[25] has provided new approaches to achieving image steganography.

We propose a novel approach which uses CNN and GAN to achieve coverless steganography. Our work makes the following contributions:

(1) We propose a method of using GAN to complete steganography tasks, whose relative payload is 2.36 bits per pixel.

(2) We propose a measurement method to evaluate the image quality of the steganography algorithm based on deep learning, which can be compared with traditional methods.

The rest of the paper is organized as follows—Section 2 briefly describes the image steganography based on GAN. We elaborate on the details of our method in Section 3. Finally, Section 4 contains our experimental results, followed by conclusions in Section 5.

2. Image Steganography Based on GAN

At present, GAN has been applied for image steganography as follows—Volkhonskiy et al. [26] first proposed the Steganographic GAN (SGAN). SGAN adopted deep GAN [27], which accounted for not only the authenticity of the generated images but also the resistance to the detection. Based on SGAN, Shi [28] proposed SSGAN . The model structure of SSGAN was similar to that of SGAN, but Wasserstein GAN [29] was adopted as the network structure, which had a faster training speed and higher image quality. The above two networks used a GAN network to generate cover images, while the Hayes GAN model proposed by Hayes et al. [21] used adversarial learning to directly generate dense images. Zhu et al. [23] put forward another method of generating hiding data with deep networks by referring to Hayes GAN's structure. It is characterized by the robustness of an adversarial sample to image changes, so that the embedded information can be extracted with high accuracy under various cover attacks (Gaussian blur, pixel loss, cropping and JPEG compression). Tang et al. [30] proposed an adaptive steganographic distortion learning framework (ASDL) to learn the cost. After several rounds of adversarial learning, the security of ASDL-GAN has been continuously enhanced, but the security has not surpassed the traditional steganic algorithm represented by S-UNIWARD [31]. Atique et al. [32] proposed another model based on an encoder-decoder to accomplish the same steganographic task and their secret images are grayscale images, but they had problems such as color distortion and poor security of secret images. Then, Hayes et al. [21] and Zhu et al. [23] made use of GAN. They used the mean squared error (MSE) for the encoder, the cross entropy loss for the discriminator, and the mean squared error for the decoder but their capacity was only limited to 0.4 bits per pixel. Zhang [33] proposed a method for hiding arbitrary binary data in images using GAN, but their experimental results were not as ideal as designed. So we are inspired by the works of Baluja and Zhang, which can improve some shortcomings.

3. Method

In general, steganography only requires two operations—encoding and decoding, consisting of three modules:

(1) An Encoder network ε, which receives a coverless image and a string of binary secret message, generates a steganographic image;
(2) A Decoder network G, which obtains a steganographic image, attempts to recover a secret message;
(3) A Discriminator network D is used to evaluate the quality of vectors and steganographic images S.

So, the architecture of our model is shown in Figure 1.

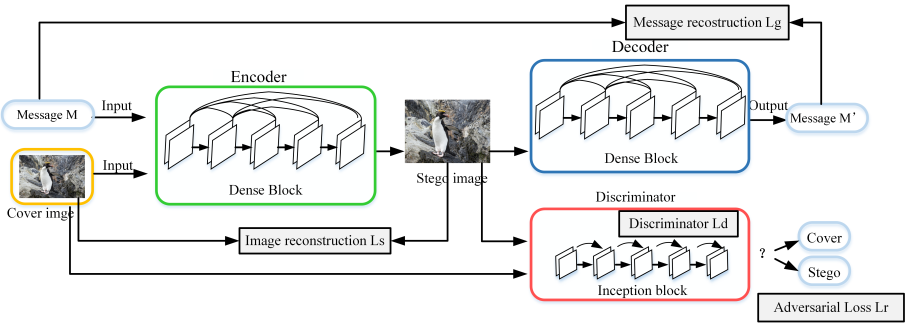

Figure 1. The architecture of the Coverless Image Steganography generative adversarial network (GAN).

3.1. Encoder Network

Firstly, we input the cover image C with the size of $(3 \times W \times H)$ and secret information M$\in \{0,1\}^{Depth \times W \times H}$ into the Encoder network ε. M is a binary data tensor of the shape $Depth \times W \times H$, where $Depth$ is the number of bits that we try to hide in each pixel of the cover image, $W \times H$ represents the size of cover images. The encoded images should look visually similar to the cover images. We perform two methods on the Encoder network ε, respectively:

(1) Use convolutional block $Conv$ to process the cover image C to get the tensor a with the size of $(32 \times W \times H)$.

$$a = \mathrm{Conv}_{3 \to 32}(C). \tag{1}$$

(2) Concatenate the message M with a and then process the tensor b with a convolutional block $Conv$. The size of b is $(32 \times W \times H)$:

$$b = \mathrm{Conv}_{32+Depth \to 32}(\mathrm{Con\,Cat}(a, M)). \tag{2}$$

Then we built two encoders models:

(i) Basic model: We apply two convolution blocks *Conv* to tensor b successively to generate steganographic images S. Formally:

$$\mathcal{E}_b(C, M) = \mathrm{Conv}_{32\to3}\left(\mathrm{Conv}_{32\to32}(b)\right). \tag{3}$$

(ii) Dense model: We use the skip connection [18] to map the features f generated by the former Dense Block to the features l generated by the latter Dense Block, as shown in Figure 1. We assume that using skip connection can improve the embedding rate. Formally:

$$\begin{cases} f = \mathrm{Conv}_{64+Depth\to32}(\mathrm{Con\,Cat}(a, b, M)) \\ l = \mathrm{Conv}_{96+Depth\to3}(\mathrm{Con\,Cat}(a, b, f, M)) \\ \mathcal{E}_l(C, M) = C + l \end{cases} \tag{4}$$

3.2. Decoder Network

The Decoder network G uses steganographic images S generated by the Encoder network ε. The Decoder network generates $M' = G(S)$, and is trying to recover the secret information tensor M according to the Reed Solomon algorithms.

$$\begin{cases} a = \mathrm{Conv}_{3\to32}(S) \\ b = \mathrm{Conv}_{32\to32}(a) \\ f = \mathrm{Conv}_{64\to32}(\mathrm{Con\,Cat}(a, b)) \\ G(S) = \mathrm{Conv}_{96\to Depth}(\mathrm{Con\,Cat}(a, b, f)) \end{cases} \tag{5}$$

3.3. Discriminator Network

In order to provide feedback on the performance of the encoder ε and generate more realistic images, we introduced a discriminator network D, which can differentiate stego images S from cover images C.

$$\begin{cases} a = \ \mathrm{Conv}_{\ 32\to32}\left(\ \mathrm{Conv}_{\ 32\to32}\left(\ \mathrm{Conv}_{\ 3\to32}(S)\right)\right) \\ \mathcal{D}(S) = \ \mathrm{Mean}\ \left(\ \mathrm{Conv}_{\ 32\to1}((a)\right) \end{cases} \tag{6}$$

XuNet, an image steganalysis, has been designed based on a CNN by Xu. For improving the statistical modeling, it embedded an absolute activation (ABS) in the first convolutional layer, and applied the TanH activation function in the shallow layers of networks to prevent overfitting, and also added batch normalization (BN) before each nonlinear activation layer. This well-designed CNN provides excellent detection performance in steganalysis. To our knowledge, it is the best-performing data-driven CNN steganalyzer based on JPEG. Therefore, we design our steganalyzer based on XuNet and adjusted it to fit our models, as shown Figure 2. The discriminator network D consists of five convolution blocks and an SPP block, and two fully connected layers with a scalar output. In order to generate scalar scores, we use the adaptive mean pool on the output of the convolution layer. In addition, we use the spatial pyramid pooling (SPP) module to replace the global average pooling layer. The spatial pyramid pooling (SPP) module [34] and its variants play a huge role in target detection and semantic segmentation models. It breaks through the limitation of fully connected layers, so that images of any size can be input to the next fully connected layers. At the same time, the SPP module can extract more features from different acceptance domains, thereby improving performance. The detailed architecture of our steganalyzer is shown in Table 1.

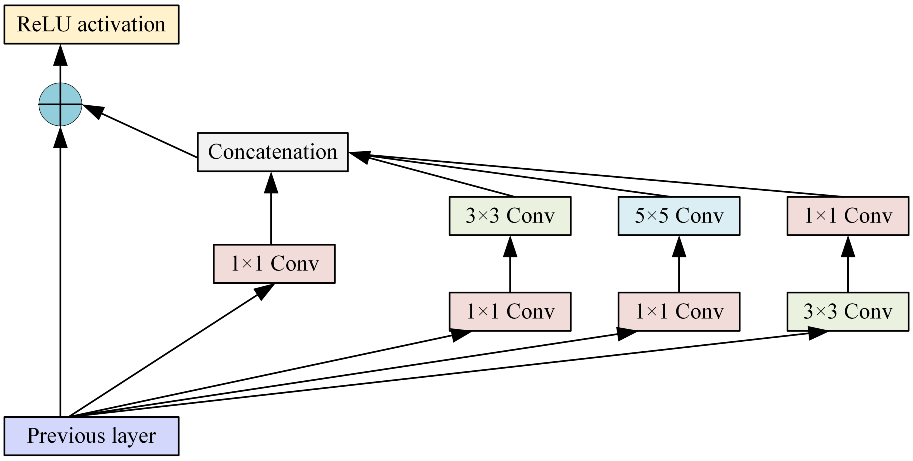

Figure 2. Inception module with residual connection.

Table 1. The architecture of the Discriminator.

Layers	Name	Output Size
Input	/	$3 \times 256 \times 256$
Layer1	ConvBlock1	$8 \times 128 \times 128$
Layer2	ConvBlock1	$8 \times 128 \times 128$
Layer3	ConvBlock2	$16 \times 64 \times 64$
Layer4	ConvBlock2	$32 \times 32 \times 32$
Layer5	ConvBlock3	$128 \times 8 \times 8$
Layer6	SPPBlock	2688×1
Layer7	FC	128×1
Layer8	FC	2×1

3.4. The Objective Fuction

c is referred to as one of the cover images C, which can be represented by the probability distribution function P. We made the cover images C follow with P and a secret message M is embedded, and the generated steganographic images S also follow the probability distribution function Q. The statistical detection ability can be quantified by the KL divergence shown in formula (7) or the JS divergence in formula (8),

$$KL(P\|Q) = \sum_{c \in C} P(c) \log \frac{P(c)}{Q(c)} \tag{7}$$

$$JS(P\|Q) = \frac{1}{2} KL\left(P\|\frac{P+Q}{2}\right) + \frac{1}{2} KL\left(Q\|\frac{P+Q}{2}\right). \tag{8}$$

The KL divergence and the JS divergence are very basic quantities, which establish the best probabilistic steganographic analysis. The original GAN’s goal is to minimize the JS divergence or the KL divergence [35]. GAN avoids the Markov chain learning mechanism in a sense, which makes it distinguishable from traditional probability generative models. Traditional probability generation models generally require Markov chain sampling and design, and GAN avoids this process with particularly high computational complexity, and directly performs sampling and correction, thereby improving the application efficiency of GAN, so its practical application scenarios are more extensive. The Encoder network ε with noise z tries to generate images which are similar with the cover images C. The Discriminator network D receives the generated images and judges them whether are the real examples or the false samples. The Discriminator network D and the Encoder network ε use cost functions (9) to play the minimax game. It trained D to maximize the probability of assigning the correct label to both training examples and samples from ε. Therefore, GAN can be used to solve the problem of steganography.

$$\text{Min}_{\varepsilon} \max_{D} V(\varepsilon, D) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_{\mathbf{z}}(z)}[\log(1 - D(\varepsilon(z)))]. \tag{9}$$

3.4.1. Encoder-Decoder Loss

In order to optimize the encoder-decoder network, this section optimizes three loss functions jointly, as shown Algorithm 1.

(1) The cross entropy loss function is used to evaluate the decoding accuracy of decoder network, that is

$$L_G = E_{X\sim p_c} \text{CrossEntropy } (G(\varepsilon(\boldsymbol{X}, \boldsymbol{M})), M). \tag{10}$$

(2) The mean square error is used to analyze the similarity between the steganographic image and the cover image, where W is the width and H is the length of image, that is

$$L_s = E_{X\sim P_C} \frac{1}{3 \times W \times H} \|X - \boldsymbol{\varepsilon}(\boldsymbol{X}, \boldsymbol{M})\|_2^2. \tag{11}$$

(3) And the realness of the steganographic image using the discriminator, that is

$$L_r = E_{X\sim P_C} D(\boldsymbol{\varepsilon}(\boldsymbol{X}, \boldsymbol{M})). \tag{12}$$

So, the training objective is to

$$\text{minimize } \left(L_G + L_s + L_r\right). \tag{13}$$

Algorithm 1 Steganographic training algorithm based on GAN

Input: Encoder ε, Decoder G, Discriminator D. threshold $_G$ ← 0.9, threshold $_D$ ← 0.85.
Output: val$_G$ ←CrossEntropy of G.

```
While val$_G$ <threshold$_G$ do
  Update $\varepsilon$ and $G$ using $L_G + L_s + L_r$.
    for $\boldsymbol{n}$ training epochs do
      if val$_G$ <threshold$_G$ then
        Update $\varepsilon$ using $L_s$, $G$ using $L_G$
      else if $val_D$ <threshold$_D$ then
      else
        Update $\varepsilon$ using $L_s + L_r$, $G$ using $L_G$
          Get $val_G$ ← CrossEntropy of $G$
          Get val$_D$ ←Cross validation accuracy of $D$
      end if
    end for
done
return $val_G$
```

3.4.2. Structural Similarity Index

Baluja [22] used the mean square error (MSE) between the pixels of the cover image and the generated image pixels as the loss function. However, MSE only penalizes the large errors of the corresponding pixels of the two images, but ignores the underlying structure of the images. Human visual systems (HVS) are more sensitive to the changes of brightness and color in textless areas, so the steganography GAN introduces the structural similarity index (SSIM) and its variant MS-SSIM [36] into the loss function.

The SSIM index compares similarity measurement tasks from three aspects—brightness δ, contrast ϵ and structure ρ. The similarity of the two images is measured by formulas (14)–(16) respectively, where μ_x and μ_y are the pixel averages of image x and image y, θ_x and θ_y are the pixel deviations of image x and image y, and θ_{xy} is the standard variance of image x and y. In addition, c_1, c_2, and c_3 are three constants to prevent the denominator from going to zero and making the formula meaningless. The general calculation method of SSIM is shown in (17), where $l > 0$, $m > 0$, $n > 0$ and they are the parameters used to adjust the relative importance of the three components. The value range of the SSIM index is [0, 1]. The higher the index, the more similar the two images. So steganography GAN uses 1-SSIM (x, y) as the loss function to measure the difference between two images. MS-SSIM is an enhanced variant of the SSIM index, so it also introduces steganography GAN's loss function.

$$\delta(x,y) = \frac{2\mu_x\mu_y + c_1}{u_x^2 + u_y^2 + c_1} \tag{14}$$

$$\epsilon(x,y) = \frac{2\theta_x\theta_y + C_2}{\theta_x^2 + \theta_y^2 + C_2} \tag{15}$$

$$\rho(x,y) = \frac{\theta_{xy} + c_3}{\theta_x\theta_y + c_3} \tag{16}$$

$$SSIM(x,y) = [\delta(x,y)]^l \cdot [\epsilon(x,y)]^m \cdot [\rho(x,y)]^n. \tag{17}$$

Considering the difference in pixel value and structure, we join MSE, SSIM and MS-SSIM together. Therefore, its mixed loss function L_D is shown:

$$L_D\left(c,c'\right) = \alpha\left(1 - \text{SSIM}\left(c,c'\right)\right) + (1-\alpha)\left(1 - \text{MS-SSIM}\left(c,c'\right)\right) + \beta\,\text{MSE}\left(c,c'\right), \tag{18}$$

where c represents the cover images, c' is the steganographic images. M is the secret message, and M' are extracted from the steganographic images. α and β are super parameters to trade off the quality of steganographic images and cover images. we set α and β of the loss function as 0.5, 0.3 respectively.

4. Experimental Results and Analysis

In this section, we will introduce our experiment details and results.

4.1. Evaluation Metrics

We take capacity, distortion, and secrecy into account. In this section, we will evaluate the performance of our model with the RS-BPP, PSNR and MS-SSIM.

4.1.1. Reed Solomon Bits Per Pixel

In the experiments, we adopt Reed-Solomon codes to accurately estimate the relative payload of our model. We call this metric the Reed-Solomon bits-per-pixel (RS-BPP), and note that it can be directly compared to traditional steganographic techniques because it represents the number of bits that are reliably transmitted in the image divided by the size of image.

4.1.2. Peak Signal-to-Noise Ratio

Peak signal-to-noise ratio (PSNR) is a commonly used image quality measurement indicator, whose purpose is to measure the distortion of the image, and has been shown to be related to the average opinion score of human experts [37].

$$MSE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (X(i,j) - Y(i,j))^2 \tag{19}$$

$$PSNR = 10 \log_{10} \left(\frac{(2^n - 1)^2}{MSE} \right). \tag{20}$$

4.2. Training

In each iteration, we match each cover image C with a data tensor M, which consists of a randomly generated sequence $Depth \times W \times H$ bits. This sequence is sampled from a Bernoulli distribution $M \sim$ Ber (0.5). In addition, we use standard data enhancement processes in preprocessing, including horizontal flipping and random cropping to the cover image C. We use the Adam optimizer with a learning rate of 1e4, normalize the gradient norm as 0.25, clip the weight of the discriminator as $[-0.1, 0.1]$, and train 32 epoch.

The experiments are conducted with the Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz, 64.00 GB RAM and one NVIDIA GeForce GTX 1080 Ti GPU.

4.3. Experimental Results

In our experiment, we used Div2k dataset (https://data.vision.ee.ethz.ch/cvl/DIV2K) to train and evaluate our model with 6 different data $Depth \in \{1, 2 \ldots, 6\}$. We used 786 pictures for training and 100 pictures for validation. Data depth means that each pixel bit of the target randomly generates data tensor shape $Depth \times W \times H$. The mean values of extracted accuracy, RS-BPP, PSNR, and MS-SSIM for the test set are recorded in Tables 2–4.

Table 2. The image quality metrics and model variant compared with Zhang's.

Dataset	Depth	Ours				Zhang's			
		Basic Model		**Dense Model**		**Basic Model**		**Dense Model**	
		PSNR	**MS-SSIM**	**PSNR**	**MS-SSIM**	**PSNR**	**MS-SSIM**	**PSNR**	**MS-SSIM**
Div2k	1	39.80	0.91	37.27	0.90	34.71	0.86	34.33	0.85
	2	36.03	0.87	36.09	0.88	34.21	0.84	34.32	0.85
	3	34.74	0.84	34.65	0.84	33.14	0.80	33.00	0.80
	4	35.59	0.86	35.35	0.85	33.73	0.83	33.99	0.83
	5	35.88	0.87	36.47	0.88	34.17	0.84	34.36	0.84
	6	36.61	0.88	36.78	0.89	34.97	0.86	34.71	0.85

Table 3. The relative payload and model variant compared with Zhang's.

Dataset	Depth	Ours		Zhang's	
		Basic Model	**Dense Model**	**Basic Model**	**Dense Model**
		RS-BPP			
Div2k	1	0.96	0.96	0.93	0.93
	2	1.82	1.83	1.76	0.93
	3	2.36	2.36	2.18	2.22
	4	2.30	2.30	2.20	2.23
	5	2.28	2.31	2.15	2.19
	6	2.24	2.27	2.17	2.18

Table 4. The accuracy of the Decoder network compared with Zhang's

Dataset	Depth	Ours		Zhang's	
		Basic Model	Dense Model	Basic model	Dense Model
		Accuracy of Recovery			
Div2k	1	0.98	0.98	0.97	0.96
	2	0.96	0.96	0.94	0.96
	3	0.89	0.89	0.86	0.87
	4	0.79	0.79	0.77	0.78
	5	0.73	0.73	0.72	0.72
	6	0.67	0.69	0.68	0.68

We randomly selected cover images to generate samples (b) (d) from the Div2k dataset. As we can see in Figure 3, steganography GAN is an efficient method which generates highly similar images according to the cover images (a) (c).

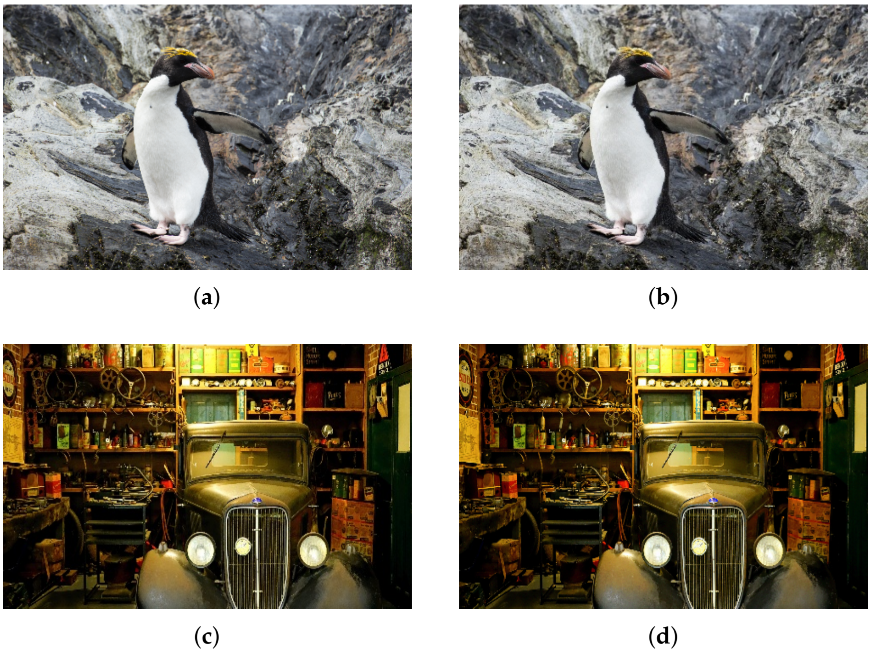
(**a**) (**b**) (**c**) (**d**)

Figure 3. The samples generated by steganography GAN. (**a**) cover image; (**b**) steganographic image; (**c**) cover image; (**d**) steganographic image.

As Tables 2 and 3 show, they are image quality measurements and the relative load of the Basic and Dense models on the Div2k dataset. In all the experiments, our model shows the best performance on almost all the indicators compared with Zhang's [33]. Focusing on the Basic model, it performs significantly well compared with the Zhang's. Table 4 shows the extracted accuracy of the Decoder network which recovers secret information. Our dense model is close to Zhang's, but the basic model behaves better.

5. Discussion and Conclusions

In this study, GAN is used to synthesize the secret information and the cover image. At this point, the secret information is embedded in any position of the composite image. On this basis, a performance index of a steganography system based on deep learning is proposed, which is convenient for direct comparison with the traditional steganography algorithm. Our models adopt different convolution methods, and the experimental results prove that our models have a high payload, the cover image especially will not be modified in the process of hiding and extracting secret information, thus ensuring the security of secret information. Furthermore, we will consider how to combine GAN with relevance feedback, compensated for the lack of user intervention, to select cover images, to increase a user's overall quality of experience. Future steps for grouping relevant items together to make the system more efficient will be investigated.

Author Contributions: Conceptualization, J.W. and J.Q.; methodology, J.W. and J.Q.; software, J.W.; validation, J.W.; formal analysis, J.W.; investigation, J.W.; resources, X.X. and H.H. ; data curation, J.W.; writing—original draft preparation, J.W.; Writing—Review & Editing, J.W., Y.T. and Z.H.; visualization, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61772561; in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ4140, 2020JJ4141; in part by the Key Research and Development Plan of Hunan Province under Grant 2018NK2012, 2019SK2022; in part by the Postgraduate Excellent teaching team Project of Hunan Province under Grant [2019]370-133; in part by the Science Research Projects of Hunan Provincial Education Department under Grant 18A174; in part by the Degree & Postgraduate Education Reform Project of Hunan Province under Grant 2019JGYB154; in part by the Postgraduate Education and Teaching Reform Project of Central South University of Forestry & Technology under Grant 2019JG013.

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

References

1. Pouli, V; Kafetzoglou, S; Tsiropoulou, E. Personalized multimedia content retrieval through relevance feedback techniques for enhanced user experience. In Proceedings of the 13th International Conference on Telecommunications, Graz, Austria, 13–15 July 2015; pp. 1–8.
2. Tan, Y; Qin, J.H.; Xiang X.Y.; Neal, N. A Robust Watermarking Scheme in YCbCr Color Space Based on Channel Coding. *IEEE Access* **2019**, *7*, 25026–25036. [CrossRef]
3. Qin, J.H.; Li, H.; Xiang, X.Y.; Tan, Y.; Neal, N.X. An Encrypted Image Retrieval Method Based on Harris Corner Optimization and LSH in Cloud Computing. *IEEE Access* **2019**, *7*, 24626–24633. [CrossRef]
4. Chen, K.; Liu, Y.J. Real-time adaptive visual secret sharing with reversibility and high capacity. *J. Real-Time Image Process.* **2019**, *16*, 871–881.
5. Bao, Z. Research on the Key Problems of Steganography in JPEG Images. Strategic Support Force Information Engineering University. 2018. Available online: https://cdmd.cnki.com.cn/Article/CDMD-91037-1018841713.htm (accessed on 20 August 2020)
6. Zhang, Y.; Luo, X.Y. Enhancing reliability and efficiency for real-time robust adaptive steganography using cyclic redundancy check codes. *J. Real-Time Image Process.* **2020**, 115–123. [CrossRef]
7. Lee, C.; Shen, J. Overlapping pixel value ordering predictor for high-capacity reversible data hiding. *J. Real-Time Image Process.* **2019**, *16*, 835–855. [CrossRef]
8. Luo, Y.J.; Qin, J.H.; Xiang, X.Y.; Tan, Y; Liu, Q. Coverless real-time image information hiding based on image block matching and Dense Convolutional Network. *J. Real-Time Image Process.* **2020**, *17*, 125–135. [CrossRef]
9. Liu Q.; Xiang X.Y.; Qin, J.H.; Tan, Y.; Tan, J.S. Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. *Knowl.-Based Syst.* **2020**, *192*, 105375–105389. [CrossRef]
10. Xiang, L.Y.; Guo, G.Q.; Yu, J.M. A convolutional neural network-based linguistic steganalysis for synonym substitution steganography. *Math. Biosci. Eng.* **2020**, *192*,11041–11058. [CrossRef]
11. Zhou, Z.L.; Sun, H.Y. Coverless Image Steganography Without Embedding. In Proceedings of the International Conference on Cloud Computing and Security, Nanjing, China, 13–15 August 2015; pp. 123–132.
12. Zheng, S.L. *Coverless Information Hiding Based on Robust Image Hashing*; Springer: Cham, Switzerland, 2017; pp. 536–547.
13. Zhou,Z.L.; Cao Y.; Sun, X.M. Coverless Information Hiding Based on Bag-of-Words Model of Image. *J. Appl. Sci.* **2016**, *34*, 527–536.
14. Otori, H. Texture Synthesis for Mobile Data Communications. *IEEE Comput. Soc.* **2009**, *29* ,74–81. [CrossRef]
15. Otori, H. *Data-Embeddable Texture Synthesis*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 146–157.
16. Wu, K. Steganography Using Reversible Texture Synthesis. *IEEE Trans. Image Process.* **2015**, *24* , 130–139. [PubMed]
17. Xu, J. Hidden message in a deformation-based texture. *Vis. Comput.* **2015**, *31*, 1653–1669. [CrossRef]

18. Huang, G; Liu, Z. Densely Connected Convolutional Networks. *arXiv* **2017**, arXiv:1608.06993v5.
19. Wang, J.; Qin, J.H. CAPTCHA recognition based on deep convolutional neural network. *Math. Biosci. Eng.* **2019**, *16*, 5851–5861. [CrossRef] [PubMed]
20. Qin, J.H.; Pan, W.Y.; Xiang, X.Y. A biological image classification method based on improved CNN. *Ecol. Inform.* **2020**, *58*, 1–8. [CrossRef]
21. Hayes, J. Generating Steganographic Images via Adversarial Training. *arXiv* **2017**, arXiv:1703.00371v3.
22. Baluja, S. Hiding Images in Plain Sight: Deep Steganography. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2069–2079.
23. Zhu, J.; Kaplan, R. HiDDeN: Hiding Data With Deep Networks. *arXiv* **2018**, arXiv:1807.09937v1.
24. Wu, P.; Yang, Y. StegNet: Mega Image Steganography Capacity with Deep Convolutional Network. *Future Internet* **2018**, *10*, 54. [CrossRef]
25. Ian, G.; Jean, P. Generative Adeversarial Nets. *NIPS* **2014**, 2672–2680.
26. Volkhonskiy, D.; Nazarov, I. Steganographic generative adversarial networks. *arXiv* **2017**, arXiv:1703.05502.
27. Radford, A. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
28. Shi, H.; Dong, J. SSGAN: secure steganography based on generative adversarial networks. In Proceedings of the Pacific Rim Conference on Multimedia, Harbin, China, 28–29 September 2017; pp. 534–544.
29. Arjovsky, M.; Chintala, S. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
30. Tang, W.; Tan, S.; Li, B. Automatic Steganographic Distortion Learning Using a Generative Adversarial Network. *IEEE Signal Process. Lett.* **2017**, *24*, 1547–1551. [CrossRef]
31. Vojtech, H.; Jessica, F.; Tomas D. Universal distortion function for steganography in an arbitrary domain. *J. Inf. Secur.* **2014**, *1*, 1–13.
32. Rahim, R.; Nadeem, M.S. End-to-end trained CNN encode-decoder networks for image steganography. *arXiv* **2017**, arXiv:1711.07201.
33. Zhang, K; Xu, L. SteganoGAN: High Capacity Image Steganography with GANs. *arXiv* **2019**, arXiv:1901.03892v2.
34. He, K.; Zhang, X.Y.; Ren, S.Q. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [CrossRef]
35. Cachin, C. An information theoretic model for steganography. *Inf. Comput.* **2004**, *192*, 41–56. [CrossRef]
36. Zhang, R.; Dong, S.; Liu, J. Invisible steganography via generative adversarial networks. *Multimed. Tools Appl.* **2019**, *78*, 8559–8575. [CrossRef]
37. Wang, Z.; Bovik, C.A. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]