electronics

Article

IDGAN: Information-Driven Generative Adversarial Network of Coverless Image Steganography

Chunying Zhang [1,2,3,4,5], Xinkai Gao [1], Xiaoxiao Liu [2,*], Wei Hou [1,2,4], Guanghui Yang [1,2,4,5], Tao Xue [1,2,4,5], Liya Wang [1,2,3,4,5] and Lu Liu [1,2]

[1] College of Science, North China University of Science and Technology, Tangshan 063210, China; hblg_zcy@126.com (C.Z.); xinkai@stu.ncst.edu.cn (X.G.); houwei@ncst.edu.cn (W.H.); yangguanghui@ncst.edu.cn (G.Y.); xuetao@ncst.edu.cn (T.X.); wang_liya@126.com (L.W.); liulu_hblg@ncst.edu.cn (L.L.)
[2] Hebei Key Laboratory of Data Science and Application, North China University of Science and Technology, Tangshan 063210, China
[3] The Key Laboratory of Engineering Computing in Tangshan City, North China University of Science and Technology, Tangshan 063210, China
[4] Hebei Engineering Research Center for the Intelligentization of Iron Ore Optimization and Ironmaking Raw Materials Preparation Processes, North China University of Science and Technology, Tangshan 063210, China
[5] Tangshan Intelligent Industry and Image Processing Technology Innovation Center, Tangshan 063210, China
* Correspondence: liuxiaoxiao@ncst.edu.cn

Abstract: Traditional image steganography techniques complete the steganography process by embedding secret information into cover images, but steganalysis tools can easily detect detectable pixel changes that lead to the leakage of confidential information. The use of a generative adversarial network (GAN) makes it possible to embed information using a combination of information and noise in generating images to achieve steganography. However, this approach is usually accompanied by issues such as poor image quality and low steganography capacity. To address these challenges, we propose a steganography model based on a novel information-driven generative adversarial network (IDGAN), which fuses a GAN, attention mechanisms, and image interpolation techniques. We introduced an attention mechanism on top of the original GAN model to improve image accuracy. In the generation model, we replaced some transposed convolution operations with image interpolation for better quality of dense images. In contrast to traditional steganographic methods, the IDGAN generates images containing confidential information without using cover images and utilizes GANs for information embedding, thus having better anti-detection capability. Moreover, the IDGAN uses an attention mechanism to improve the image details and clarity and optimizes the steganography effect through an image interpolation algorithm. Experimental results demonstrate that the IDGAN achieves an accuracy of 99.4%, 95.4%, 93.2%, and 100% on the MNIST, Intel Image Classification, Flowers, and Face datasets, respectively, with an embedding rate of 0.17 bpp. The model effectively protects confidential information while maintaining high image quality.

Keywords: coverless steganography; generative adversarial network; attention mechanisms; image interpolation; dense convolutional network

check for updates

Citation: Zhang, C.; Gao, X.; Liu, X.; Hou, W.; Yang, G.; Xue, T.; Wang, L.; Liu, L. IDGAN: Information-Driven Generative Adversarial Network of Coverless Image Steganography. *Electronics* **2023**, *12*, 2881. https://doi.org/10.3390/electronics12132881

Academic Editor: Yung-Lyul Lee

Received: 28 April 2023
Revised: 31 May 2023
Accepted: 27 June 2023
Published: 29 June 2023

1. Introduction

With the development of internet technology, digital communication has been widely used for multimedia data transmission. At present, the internet produces a massive amount of multimedia information every day. Therefore, ensuring the secure transmission and storage of data has become a fundamental task of communication. Data security and privacy protection have become one of the important global concerns. Steganography, as an important information-hiding technology, has become one of the fields extensively studied by experts and scholars both domestically and internationally. By using brand new information storage methods during the process of information transmission, steganography can not only protect information security but also ensure behavior security.

Traditional image steganography is usually divided into transform domain steganography [1] and spatial domain steganography [2,3]. Transform domain steganography improves the detection-resistant ability of steganographic images by modifying the cover image. Based on spatial domain steganographic algorithms ranging from the original least significant bit (LSB) method, pixel-value differencing (PVD) method to the highly undetectable stego (HUGO) method [4], adaptive steganography has now covered a wide range of content, including wavelet obtained weight (WOW) and spatial universal wavelet relative distortion (S-UNIWARD) approaches.

With the rapid development of neural networks in the field of computer vision, image steganography began utilizing the powerful learning ability of neural networks to help to find the most suitable embedding positions in the image cover, forming a visually imperceptible secret cover for covert communication. Research on high-capacity image steganography based on neural networks quickly emerged. Baluja from Google Research Institute [5] was the first to embed an equally sized color secret image in a color image, which was first processed by a pre-processing network, and then together with the cover image was encoded by the encoding network to generate a secret image that resembled the cover image. During the recovery process, the secret image with high fidelity could be restored by passing the secret image to the decoding network, and the amount of secret information embedded reached 24 bpp. At the same time, Rahim et al. [6] proposed a neural network model based on the encoding and decoding concept, which embedded a gray image in a color image while maintaining a high accuracy of secret image restoration. However, high-capacity image steganography based on neural networks is still in its infancy. The visual quality of the secret images still has a lot of room for improvement. Moreover, most secret images that undergo embedding operations are vulnerable to security analysis. The use of residual images to enhance the difference between the original cover image and the secret image has led to cases of secret information leakage. Therefore, although neural network technology is widely used in other computer vision fields, research on high-capacity image steganography supported by neural networks is still in its early stages.

In the field of coverless image steganography, Volkhonskiy et al. [7] proposed the first GAN-based image steganography model, SGAN. This model first uses a DCGAN [8] to transform random noise into cover images and then uses traditional embedding algorithms to embed secret information into generated cover images to generate covert images. Since the DCGAN used had instability in the training process, the quality of the generated cover image could not meet the transparency requirements of the steganography algorithm. Therefore, Shi et al. [9] replaced the DCGAN used in the SGAN with a WGAN [10] and proposed a SSGAN. Compared with SGANs, cover images generated by SSGANs are visually closer to real images and to some extent avoid the instability during model training. Subsequently, Wang et al. [11] further optimized the model framework based on a SGAN and SSGAN, and proposed a Stego-WGAN. The biggest difference from the SGAN and SSGAN is that the Stego-WGAN takes the covert image and the original image as inputs of the discriminative network, which not only ensures the generated image for embedding secret information but also maintains the visual consistency of the covert image and the original image to some extent. In order to further improve the transparency of the algorithm, researchers used neural networks in the design of the embedding distortion function. Tang et al. [12] combined GAN and STC [13] encoding to propose the ASDL-GAN. Experimental results show that the performance of the ASDL-GAN still has a certain gap compared with traditional adaptive steganography methods, and the use of the Ternary Embedding Simulator activation function in the model increases the training cycle of the model, resulting in a much longer training time than traditional steganography algorithms. To improve the performance of the model and reduce training time, Yang et al. [14] replaced the Ternary Embedding Simulator function in the ASDL-GAN with a Tanh simulator and proposed the UT-SCA-GAN model. Fan et al. [15] learned an image steganography scheme represented by a restricted neural encoder through constructing a restrictive neural encoder and an adversarial model, AdvSGAN. Zheng et al. [16] proposed a robust image hash steganography algorithm that not only increases steganography capacity but also effectively reduces the size of local image library through reordering. Hu et al. [17] constructed the first non-embedding steganography method based on a GAN network using the DCGAN model. In this method, secret information is used to directly generate realistic stego images, and a CNN-based extraction model is employed to recover the secret information. Most of the aforementioned schemes only considered the extent to which the generated images can carry secret information, without taking into account the quality of the generated images and the resistance to analysis capability of the covert images.

This paper proposes an IDGAN model that utilizes improved generative adversarial network techniques to implement information-driven steganography via image interpolation. To evaluate the effectiveness of the IDGAN, this study tests the model on four datasets, including MNIST, Intel Image Classification, Flowers, and a large self-constructed dataset called Face. The first three datasets are public, and the fourth one contains high-quality human facial portraits with multiple semantic elements, such as age, skin tone, background, and decorations. The last dataset allows a comprehensive evaluation of the model's capability to exclude image outliers from the training data. The principal contributions of this paper are as follows:

1. The IDGAN model is proposed, utilizing a fusion attention mechanism and image interpolation to perform coverless steganography tasks, achieving a steganographic embedding rate of 0.17 bpp.
2. Compared with other coverless steganography techniques, the IDGAN incorporates a complete information-driven network structure for generating covert images while implementing image interpolation and attention mechanisms to enhance image quality.
3. In terms of its anti-analysis ability, the IDGAN model greatly surpassed traditional image steganography methods by utilizing an information-driven approach for generating covert images.

The outlined structure of this paper consists of five parts. Section 2 provides a review of current steganography techniques based on a generative adversarial network (GAN). In Section 3, we perform a detailed examination of the IDGAN model. Section 4 provides an analysis of our experimental results, and finally, we present our conclusions and potential future directions in Sections 5 and 6.

2. Related Work

Compared to the traditional image steganography techniques, embedding and extracting information without prior knowledge were made possible by using encoding-decoding network for the fusion of image–text and image–image. Hayes et al. [18] proposed the HayesGAN for embedding text information in images using an encoding network to fuse the secret message with the carrier image to generate a covert image, followed by a decoding network for message retrieval. However, this type of model finds it difficult to extract hidden information when images are subjected to noisy attacks during practical application of the technique. Therefore, the robustness of the model has become an essential criterion for evaluating this approach. To enhance robustness, Zhu et al. [19] proposed the HiDDeN image steganography framework, which includes a noise layer between the encoding and decoding networks to simulate possible noisy attacks in images and increase the model's resistance to noise attacks. However, the design of the network structure resulted in limited embedding capacity for the model. As embedding capacity increased, the accuracy of information extraction gradually decreased, leaving room for further improvement in the model's robustness. Wu et al. [20] designed the StegNet model, which uses a loss function consisting of L1 norm and variance. Duan et al. [21] improved this model by combining the U-Net [22] network with a similar encoding network structure to improve the quality of generated images. Fu et al. [23] also used a similar U-Net encoding structure and combined the GAN ideas to propose the HISGAN. In 2021, Mo et al. [24] proposed the MDRSteg framework based on the multiscale fusion residual network and hollow convolution, achieving high-capacity image steganography. Lu et al. [25] proposed a reversible steganography network for high-capacity image steganography, where the model considers the covert and secret images as a pair of inverse problems in the image transformation domain and uses forward and backward propagation of a single reversible network to embed and extract information. Rahim et al. [6] took a different approach by embedding single-channel grayscale images into three-channel colored images. However, the model's performance was poor due to the resulting colored distortion in the covert image. Furthermore, most image steganography methods based on covert image generation focuses only on the visual similarity between the covert and carry images. They do not consider whether the images can resist steganalysis algorithm attacks, resulting in weak steganography analysis detection capabilities. Zhang et al. [26] resolved this issue by taking advantage of the Y channel color space in YUV, which does not contain color information. They embedded grayscale images in the Y channel to address the problem of color distortion in covert images. Additionally, the model uses the advanced steganalysis network XuNet [27] as a discriminator network to improve the generated covert image's resistance to steganalysis detection through continuous adversarial training between the discriminator and generator networks. Table 1 summarizes the encoding-decoding-network-based steganography mentioned above.

Table 1. Encoding-decoding steganography model.

Model	Characteristics
HayesGAN	Uses encoding-decoding network to hide text information in images but is sensitive to noise
HiDDeN	Adds noise layer between encoding-decoding network to improve anti-noise-attack ability
StegNet	Uses L1 norm- and variance-based loss function to improve performance
HISGAN	Combines a GAN with similar encoding structures into U-Net, improves image quality and embedding capacity
MDRSteg	Implements high capacity image steganography based on dilation convolutions and multiscale residual network
Reversible Steganography Network	Considers embedding and extracting secret images as an inverse problem and uses single reversible network
"Gray Hiding in Color"	Poor model performance due to color distortion
"Gray Giding in Y"	Embeds grayscale images into Y channel to avoid color distortion, improves security using advanced steganalysis network

Although image steganography methods based on covert image generation show good performance, the existing models still face challenges such as poor quality of generated covert images, significant differences between decrypted and original images, and more crucially, inadequate consideration of the models' security and robustness. As a result, most of the work is insufficient in terms of the model's resistance to steganalysis detection. To address these issues, this study builds upon such methods and uses a GAN, attention mechanisms, and image interpolation for image steganography research.

2.1. Generative Adversarial Network

A generative adversarial network (GAN) is a new network framework proposed by Goodfellow et al. [28] in 2014. A GAN uses a "zero-sum game" between two networks to learn from each other and finally transforms random noise into fake data that can be indistinguishable from the real ones. The model framework is shown in Figure 1.

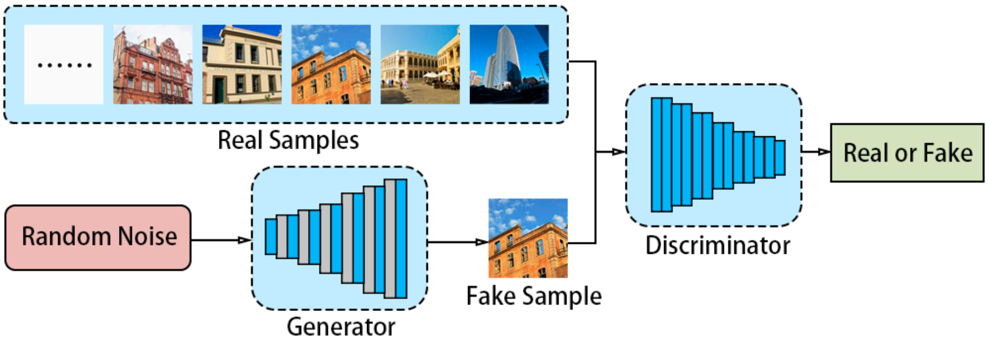

Figure 1. Structure of the generative adversarial network.

The training process of a GAN can be summarized as a minimax game. The discriminative network, D, needs to distinguish whether the input data are real or fake as accurately as possible. The optimization process can be represented by the following Formula (1):

$$\max_D V(D,G) = E_{x \sim p_{data}}[\log(D(x))] + E_{z \sim p_z}[\log(1 - D(G(z)))] \tag{1}$$

where P_{data} represents the distribution of real data, P_z represents the distribution of noise, E represents expectation, $D(x)$ represents the probability determined by the discriminator network D when outputting the real data x, $G(z)$ represents the generator, and $D(G(z))$ represents the probability determined by the discriminator network D when outputting the generated data.

The generative network, G, aims to make it impossible for the discriminative network, D, to distinguish whether the input data are real or fake as much as possible. This process can be described by Formula (2) as follows:

$$\max_G V(D,G) = E_{z \sim p_z}[\log(D(G(z)))] \tag{2}$$

The overall objective function of the original GAN network is shown in Formula (3):

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log(D(x))] + E_{z \sim p_{data}(z)}[\log(1 - D(G(z)))] \tag{3}$$

2.2. Self-Attention Mechanism

Attention mechanisms can be described as a combination of functions that report a reasonable intensity of attention when querying for a particular pixel position. The function outputs a weighted sum of the key-value pairs of the query object, and the weight assigned to each value is computed with the query-key pair. Vectors serve as the input for the three parts of the attention mechanism. Attention intensities can be computed using cosine similarity, perceptrons, dot-product scalars, among other methods. This study employs attention mechanisms as self-attention models, a variant of attention mechanisms that reduces dependence on external information while emphasizing the search for internal feature correlations within an image. Queries Q(query), keys K(key), and values V(value) are all automatically generated using different computation methods based on the image or feature itself. The attention intensities are computed using dot-product scalars and yield scalar output. Figure 2 shows the attention model utilized in this study.

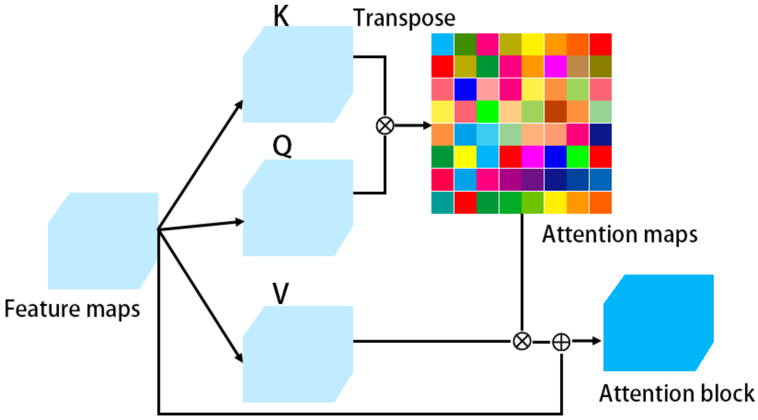

Figure 2. Diagram of the attention module structure.

The matrices $K_{n\times c}$, $Q_{n\times c}$ and $V_{n\times c}$ are three identically sized parameter matrices obtained from the convolutional output of the same feature map. Here, n represents the number of image pixels, and c represents the number of channels in the image. The row vector of Q represents the feature of the pixel channel to be queried, while all column vectors of K^T represent the channel features of all pixels. V represents the feature map of the original image. The calculation process follows Formula (4) below.

$$\begin{cases} s_{i,j} = Q \cdot K^T \\ p_{i,j} = \frac{\exp(s_{i,j})}{\Sigma_{k=1}^{n} s_{i,k}} \\ a_{i,j} = \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} \Sigma_{k=1}^{n} p_{i,k} \cdot V_{k,j} \end{cases} \quad (4)$$

First, the row vector of Q and all column vectors of K^T are dot-produced to obtain the similarity score $S_{n\times n}$ between the current query position and all image pixels, representing the global correlation features of image pixels. Second, the similarity score $S_{n\times n}$ is normalized by rows to obtain a probability distribution of similarity $p_{n\times n}$. Finally, the dot product of the probability distribution of image similarity and the feature map of the original image is used to obtain the final attention intensity of the image pixel, $a_{i,j}$. The working mechanism of the self-attention module in this study can be expressed as the following simplified Formula (5):

$$\begin{cases} X = input \\ out = X + \omega \times soft\max(X \cdot X^T) \cdot X \end{cases} \quad (5)$$

Here, X represents the input part of the attention module, and ω is an adaptive parameter that can be trained. However, there is no distinction between query and key-value components because they all come from the input, X. As can be seen from the simplified formula, the attention module does not introduce additional information but rather captures global image features solely through self-variant operations.

2.3. Image Interpolation

Image interpolation refers to the process of generating new pixels through algorithms based on existing image pixels to increase the image resolution or change the image size. The nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation are three commonly used image interpolation algorithms. In an IDGAN, bilinear interpolation, which has moderate complexity and good results, is selected for image interpolation. For an image I, if it needs to be interpolated into a new image I_{new} of $H \times W$ (H and W represent the height and width of the interpolated image), the interpolation process can be described as in Formula (6):

$$I_{new}(i,j) = \sum_{m=0}^{1} \sum_{n=0}^{1} w_{m,n} I(i+m, j+n) \quad (6)$$

Here, $w_{m,n}$ represents the weight coefficient used in bilinear interpolation, and it is determined by computing the distance between the interpolation position and the four closest pixels. Bilinear interpolation allows for the scaling of an image to any size, which is essential for embedding ciphertext in this study.

During the design of the generative models, we carried out an analysis of the Face dataset to identify the causes of poor image quality resulting from the deconvolution network. We found that during the initial training phase, the use of the deconvolution method for image generation resulted in the formation of a regular grid or block-like patterns. However, using the bilinear interpolation method rapidly produced initial facial contours and significant facial features. The comparison of two images in Figure 3 depicts the blurred image of the image generation at its initial training stage. It is evident that the generative models that use bilinear interpolation have the ability to efficiently avoid the occurrence of blurred and interlaced pixels in the generated images.

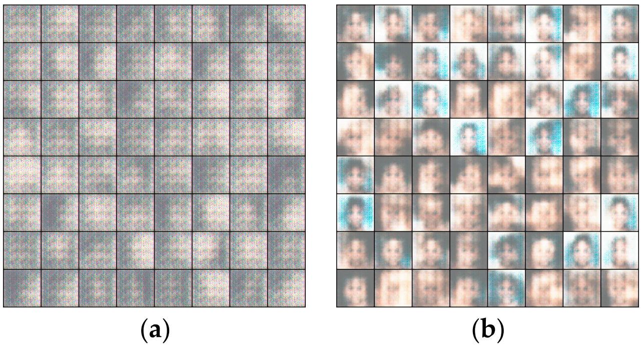

Figure 3. Comparison of deconvolution and image interpolation: (**a**) deconvolution and (**b**) bilinear interpolation.

After comparing the results, we replaced some of the deconvolution operations in the generative network model with bilinear interpolation.

3. IDGAN Steganography Model

This paper proposes an IDGAN (an information-driven generative adversarial Network), an information-driven steganography model based on attention mechanisms and image interpolation techniques, to address the issue of low information recovery accuracy and poor image quality in current coverless image steganography. To minimize information loss, the IDGAN employs the Hamming code and DenseNet structures. By combining the generative, extraction, and discriminative models, this model can generate naturally concealed images from any data. The IDGAN model structure shown in Figure 4 is comprised of three parts.

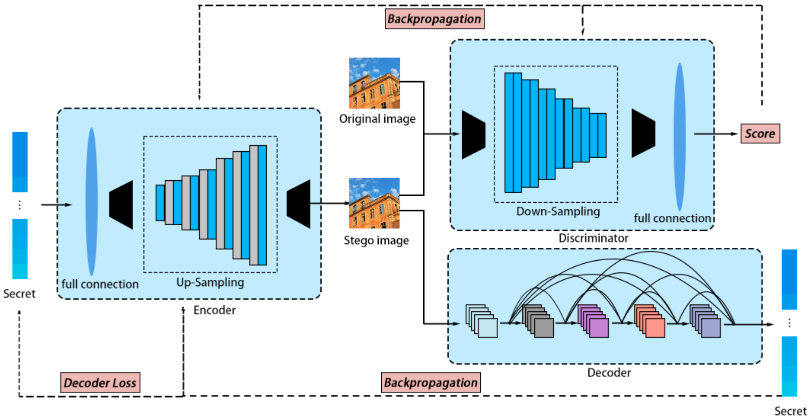

Figure 4. IDGAN model structure.

The proposed model consists of three modules, namely the generative G, discriminative D, and extraction E models. The generative model receives preprocessed binary secret messages to generate a concealed image. The discriminative model evaluates the quality of the concealed image, while the extraction model accepts the quantized steganographic image and extracts the secret message through feature extraction.

3.1. Generative Model

After concatenating the secret information, it enters the generative model G and undergoes a linear transformation through a fully connected layer. The pixel features are folded and arranged in the general form of the image pixels $H \times W \times C$. Here, $H \times W$ represents the dimensions of the feature map, and C represents the number of image channel features. Subsequently, the image is pixel expanded using bilinear interpolation combined with a convolutional neural network for upsampling, which performs better than the deconvolution method. The initial pixel feature is 4 × 4 × 512, but after bilinear interpolation, it becomes 8 × 8 × 512. The model then utilizes an Up Block module to learn more detailed features using convolutional layers to cooperatively upsample it. To capture a larger range of image correlations and avoid the limitations of the convolutional kernel receptive field, this method employs attention mechanisms. The model passes the image feature channel information through the attention module to learn the primary global contours and detailed features, and then performs interpolation and feature extraction. The resulting feature map is converted to 32 × 32 × 64, and the attention module is used again for global semantic feature learning, reducing image distortions and background anomalies. The final concealed image is 64 × 64 × 3 in size. The complete structure of the generative model is shown in Figure 5.

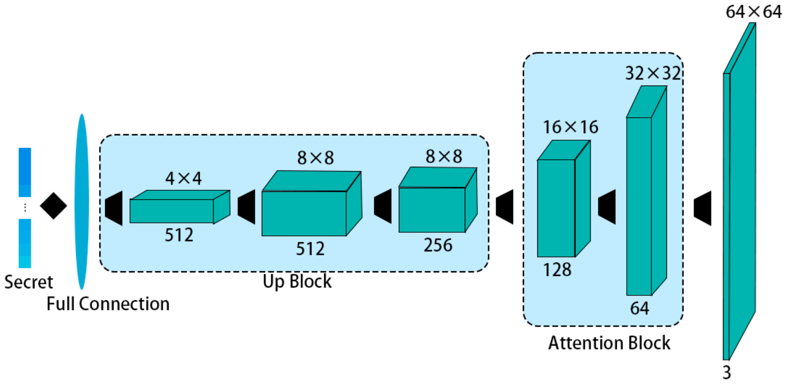

Figure 5. Generative model structure.

3.2. Discriminative Model

The quality of the discriminative model in the system directly influences the quality of the entire image generation process and impacts the security and reliability of the steganography algorithm. The discriminative model's network structure is comparable to that of the DCGAN model, with a stack of convolutional and fully connected layers, except for two layers being substituted by the previously introduced attention module. The discriminative model must correctly learn the pertinent image features for the propagation of the accurate error back to the generative model. Despite the DCGAN discriminator's competent architecture, the performance of its loss function remains unimpressive. The DCGAN uses Jensen–Shannon divergence to measure the similarity between probability distributions of two datasets. The Jensen–Shannon divergence is the symmetric form of Kullback–Leibler divergence, as shown in Formulas (7) and (8):

$$KL(P \parallel Q) = \Sigma_{c e C} P(c) \log \frac{P(c)}{Q(c)} \tag{7}$$

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

The two loss functions, in essence, calculate the relative entropy between distinct probability distributions. The KL divergence value loses its significance when there is a complete absence of overlap between two data distributions, as opposed to the JS divergence value which is computed as a constant under these circumstances. This implies a zero gradient in deep learning models. A mere shift in the two distributions without any overlap makes the model challenging to train successfully. The Wasserstein distance model is often employed in non-embedded steganography that uses generative adversarial networks to enhance image quality. The feature of the Wasserstein distance is its calculation based on the genuine difference in distance between the distinct distributions. It can still result in an appropriate gradient calculation and loss value, even when there is no intersection between the two data distributions. As shown in Formula (9):

$$W\left(P_r, P_g\right) = \underset{\gamma\epsilon\Pi\left(P_r,P_g\right)}{inf} \; E_{(x,y)\sim}[\parallel x - y \parallel] \tag{9}$$

In this, γ represents the joint distribution of the real image and the generated image. In practice, due to the strong constraint ability of the WGAN loss function on image authenticity, it is difficult to extract the secret message completely. Therefore, it needs to be used in conjunction with the JS divergence, and trained in stages with different weight coefficients, which is more cumbersome and inefficient. Therefore, this chapter adopts a completely different discriminator loss function design. Based on the characteristics and actual performance of the soft margin model HingeLoss, the IDGAN model adopts a loss function that can both provide correct gradients for parameter training and be combined with image steganography. This can be seen in Formula (10) as follows:

$$\min_{G} \max_{D} V(G,D) = E_{x-p_{data}(x),z-p_z(z)}[\max(0, 1-(D(x)-D(G(z)))] \tag{10}$$

The formula indicates that the discriminator model can identify differences between authentic and synthetic data from different perspectives. The generator needs to minimize these differences. Figure 6 illustrates the complete structure of the discriminator model.

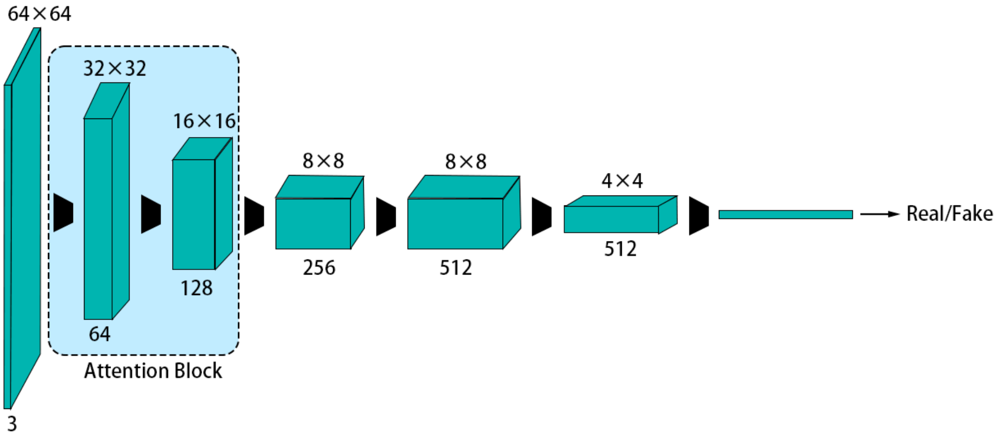

Figure 6. Discriminator model structure.

Unlike the traditional DCGAN, which uses the JS divergence as a loss function, our model employs a HingeLoss based on the soft margin model. This ensures that the correct gradients are provided for parameter training and is consistent with the generator model in the IDGAN. Our model searches for differences between real and generated samples from different perspectives, thereby enabling the generator to minimize these differences.

3.3. Extraction Model

After receiving the steganographic image, the extraction model continuously extracts features to recover the secret message embedded in it. The network structure and loss function design have two important directions: (1) complete recovery of the secret message and (2) minimal interference with the image generation process.

The IDGAN adopts a DenseNet for its network structure, allowing deep neural networks to use shallow feature maps for better extraction performance. Additionally, a feature extraction layer with an attention mechanism is used between shallow, middle, and deep layers. With a global approach to identifying feature correlations, the extractor can combine the secret message features with the image features to facilitate effective collaborative training of the generator and extraction models. The network structure of the extraction model is presented in Figure 7.

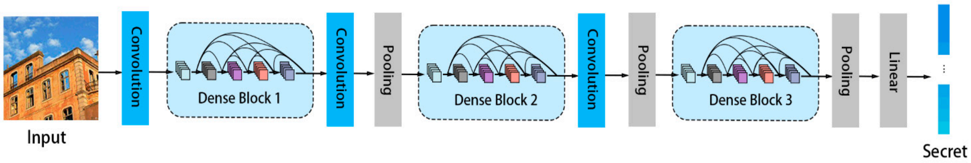

Figure 7. Extraction model structure.

While mean square error (MSE) functions have been shown to be effective for image reconstruction and information recovery, the steganography model constructed in this paper requires simultaneous training of the extractor and generator. The generator parameters need to optimize and learn from two loss functions in different directions. Our experiments show that the extractor loss decreases rapidly in the early stage, improving the accuracy of information recovery significantly. At this point, however, the generated images did not yet express specific dataset features. The rates of steganography and image generation training differ greatly. Prior work imposed constraints by manually adjusting training phase weights. In this section, we require an extractor loss function to control the gradient size during the training process and ensure complete information recovery. The loss function used in this section is shown in Formula (11):

$$L_{G,E} = \min E_{z\sim p.(z)}\left[\max(0,(z-E(z)))^2-\alpha\right]\times\beta \tag{11}$$

Parameter α is the threshold that controls the prescribed accuracy. The optimization process of the extractor and the generator must stop once the information recovery accuracy reaches the required level to prevent unnecessary parameter updates affecting image generation. Parameter β exclusively weights the loss function of the extractor to constrain the training gradient size, thereby preventing training from becoming too fast.

4. Experiment and Results

4.1. Experimental Environment and Dataset

4.1.1. Experimental Environment

The experiment employed an Intel Xeon Gold 5218 CPU, equipped with 64 GB of memory. The graphics processing unit (GPU) utilized was an Nvidia RTX3070Ti. Additionally, the experiment ran on Ubuntu 20.04 and was completed using the PyTorch framework.

4.1.2. Dataset

For the purpose of experimentation, the proposed model in this paper was tested using four datasets, which were either selected or created. Three of these datasets, namely MNIST, Intel Image Classification, and Flowers datasets, are publicly available. Besides acknowledging the limitation of using a single dataset for image generation, we constructed a large-scale, realistic facial dataset in this study. The dataset is built on high-quality human facial portraits, and it has diverse features, such as different ages, skin colors, background environments, and decorative items. These features were included to test whether the developed model is capable of eliminating the characteristics of large outliers in the image data. The MNIST dataset comprises handwritten images of digits from 0 to 9, with a training set of 60,000 grayscale images and a test set of 10,000 images. Each image is 28 × 28 pixels in size. The Intel Image Classification dataset contains six categories. For this experiment, we chose 3000 building images from the first category, and each image has a size of 150 × 150 pixels. The Flowers dataset comprises different flower categories. For this experiment, we selected 900 images of flowers from the third category, with varying image sizes. Moreover, the Face dataset that we constructed for this paper contains features such as hair, skin, background, and decorations. Specifically, it includes 20,000 facial images of size 256 × 256 pixels. Figure 8 is a partial display of the dataset.

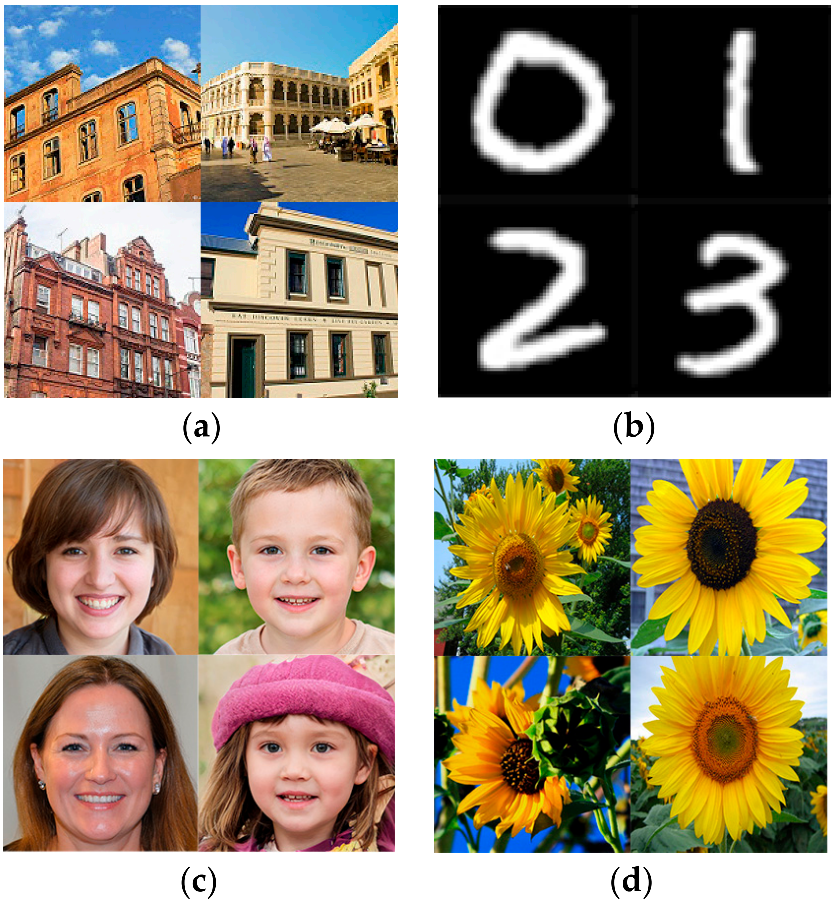

Figure 8. Partial display of the dataset: (**a**) images from Intel Image Classification, (**b**) MNIST, (**c**) Face and, (**d**) Flowers datasets.

4.1.3. Data Processing

To convert encrypted information into binary form and ensure uniformity of the information format, it is necessary to pad incomplete information. Hamming code can be used to encode the information and map the encoded binary information randomly to a range of [−700, 700], which improves information confidentiality. Our experimentation confirms that this coding method improves accuracy of information recovery. Additionally, normalizing the mapped information can help to mitigate the problems of vanishing and exploding gradients during training, and aid in parameter convergence and image synthesis. Finally, the data are expanded to a high-dimension vector to meet the model's input requirements. This preprocessing step provides the model with effective and reliable input data, thereby enhancing the model's performance. The specific information processing flow is shown in Figure 9.

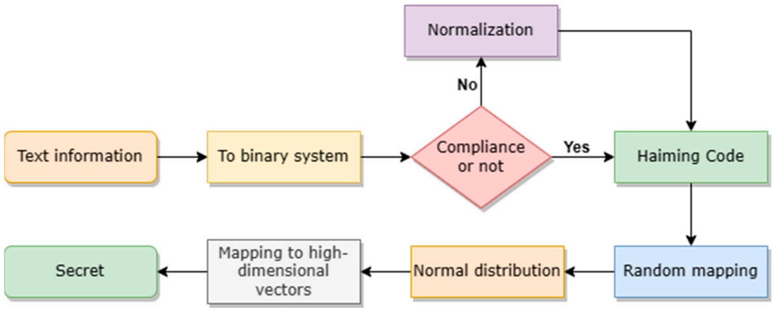

Figure 9. Information processing pipeline.

4.2. Training Process

The IDGAN model's structure employs binary cross entropy with 0–1 classification and the binary cross-entropy function for the optimizer of both the discriminator D and the generator G. The confidential information produced by both the discriminator D and the information extractor E must undergo scaling to values ranging from 0 to 1 through the sigmoid function. The Formula is presented as follows (12):

$$Loss(X_i, y_i) = -\omega[y_i \log x_i + (1 - y_i)\log(1 - x_i)] \tag{12}$$

During the training of the IDGAN model, the discriminator inputs every training sample and outputs a scalar value X_i, which represents the probability that the sample is real. X_i is calculated based on the difference between the input sample and the true label y_i. The generative model's loss function is computed through the weighted summation of the probability that the generated images are real and the image reconstruction loss using the binary cross-entropy function. Backpropagation is utilized to update the weight parameters of the generative model, with the aim of maximizing the probability that the discriminator judges the generated image as real and minimizing the image reconstruction loss. The discriminator's loss function uses the binary cross-entropy function and includes labels for both real (1) and fake (0) samples. In the IDGAN training scheme, both the undisguised real images and the disguised images generated by the generative model are input into the discriminator to calculate their losses. After the loss function gradient is returned to update the generative model's parameters, it is then passed back to the discriminator to update its parameters to better discriminate between real and disguised images. The secret information loss value is computed using the MSE function, which calculates the root mean square between the confidential data output and the bitstream value of input G, as shown in Formula (13):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \overset{\Lambda}{Y}_i\right)^2 \tag{13}$$

$\overset{\Lambda}{Y}_i$ represents the secret information obtained by the extraction model, and $\overset{\Lambda}{Y}_i$ represents the true secret information. During the training process of the IDGAN model, the value of MSE is computed for every sample, and the gradient is calculated through backpropagation to update the parameters of both the generative and extraction models. Every iteration aims at minimizing the MSE value, enabling the extraction model to learn the correct features and improving the accuracy of information extraction.

The training and optimization process of the IDGAN network involves the use of the Adam optimizer with a learning rate of lr = 0.0002 and a batch size of 64. Each batch consists of data passed through a concatenated cycle of D, G, and E and trained for one iteration. For each iteration, 64 sets of information $\{m_1, m_2, \ldots\ldots, m_{720}\}$ are used as inputs for the generative model. The gradients of D are calculated using Formula (13) and the shared convolution layer parameters are updated according to the Adam rule.

$$\nabla_{\theta_d} = \frac{1}{m}\sum_{i=1}^{m}[\lg D(x_i) + \lg(1 - D(G(z_i, y_i, d_i)))] \tag{14}$$

The gradient of G is calculated based on Formula (14), and updated using the Adam optimizer to update the parameters of G.

$$\nabla_{\theta_G} = \frac{1}{m}\sum_{i=1}^{m}\lg D(G(z_i, y_i, d_i)) \tag{15}$$

An iterator is formed by concatenating the parameters of network G with the shared convolutional layer, and gradients are computed based on Formula (15). Then, the Adam optimizer is used to update the parameters of both G and all shared convolutional layers.

$$\nabla_{\theta_{G,S}} = \frac{1}{m}\sum_{i=1}^{m}CE(y_i|G(z_i, y_i, d_i), y_i) + MSE(d_i|G(z_i, y_i d_i), d_i) \tag{16}$$

The algorithm flow is as Algorithm 1.

Algorithm 1. Algorithm Training Process

```
Input: Secret information $S$
Output: Encrypted images $C$, Decrypted information $S'$
# Parameter
n = epoch;
m = number of training sets;
b = batch size;
# Initialize three networks
Generator = initialize_Generator();
Discriminator = initialize_Discriminator();
Decoder = initialize_ Decoder ();
# Training process
for i = 1 to n do:
    for j = 1 to m/b do:
        # Generate encrypted images
        S_batch = randomly_select(S, b);
        Encrypted_images = Generator(S_batch);
        # Train Discriminator
        Discriminator_loss = train_Discriminator(Discriminator, Encrypted_images, b)
        # Train Generator
        Generator_loss = train_Generator(Generator, Discriminator, S_batch, b)
        # Train Decoder
        Decoder_loss = train_Decoder(Decoder, Generator, S_batch, b);
    # Decode secret information
    Decrypted_info = Decoder(Encrypted_images);
    # Calculate errors
    Image_error = calculate_error(Real_images, Encrypted_images);
    Info_error = calculate_error(S, Decrypted_info);
# Update network parameters
    Generator.update(Generator_loss);
    Discriminator.update(Discriminator_loss);
    Decoder.update(Decoder_loss);
return Encrypted images $C$, Decrypted information $S'$.
```

4.3. Experimental Results

4.3.1. Evaluation Metrics

In this experiment, three metrics were used to evaluate the performance of the model: information extraction rate, FID evaluation, and steganography capacity.

The information extraction rate is a crucial metric for evaluating the steganography performance of the model, as it provides an assessment of the model's ability to extract secret information from steganographic images with accuracy. Its calculation method is as follows: $IER = \frac{\overset{\Lambda}{M_i}}{M_i}$. It figures out the ratio of the correctly extracted secret information to all embedded secret information. The term "correctly extracted secret information" indicates the information extracted is identical to the original secret information. In the experiment, we generated steganographic images by randomly generating secret information, which was then embedded into cover images. $\overset{\Lambda}{M_i}$ denotes the extracted information after steganography, while M_i represents the original information before steganography.

To evaluate the quality of the generated steganographic images, we used the FID metric, which measures the similarity between the generated and real images. FID calculation involves computing the Fréchet distance between the feature vectors extracted from the generated and real images using a pre-trained deep convolutional neural network. A lower FID value indicates a smaller difference between the distributions of the feature vectors, implying higher image similarity. Formula (17) shows the formula for FID evaluation.

$$FID = \left|\left|\mu_r - \mu_g\right|\right|^2 + Tr\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r\Sigma_g\right)^{1/2}\right) \tag{17}$$

In this research, we adopt several variables. Value μ_r represents the feature mean value of genuine samples, while μ_g represents the feature mean value of generated samples that contain secret information. The covariance matrix of genuine samples is represented by Σ_r and the covariance matrix of generated samples that contain secret information is represented by Σ_g.

Bits per pixel (bpp) is a crucial metric to measure steganography capacity in digital image steganography. Its calculation method is as follows: $bpp = \frac{bits}{pixel}$, where bpp specifies the number of bits that can be embedded into each pixel. The steganography capacity, which is determined by the number of bpp, represents the maximum amount of hidden information that can be concealed within a carrier image, and is regarded as an important factor to assess the feasibility of steganography algorithms.

We conducted an image size analysis across varying lengths of secret information that were embedded. We performed this through previous experiments and our customized strategy in order to calculate the steganography capacity index, a critical indicator of our study. We comprehensively assessed the performance of our model through three metrics, namely the information extraction rate, FID evaluation, and steganography capacity. These assessments provide a reference point for optimizing and designing new steganography models.

4.3.2. Extraction Efficiency

We conducted comparative experiments on four datasets, MNIST, Intel Image Classification, Flowers, and Face. The results suggest that these datasets achieve satisfactory accuracy rates when the embedding dimension was 600. Subsequently, we increased the information dimension and evaluated the information accuracy rates of buried information in varying dimensions, 600, 720, and 840. We visually present a subset of these datasets containing hidden information in Figure 10, where an embedding dimension of 600 was used.

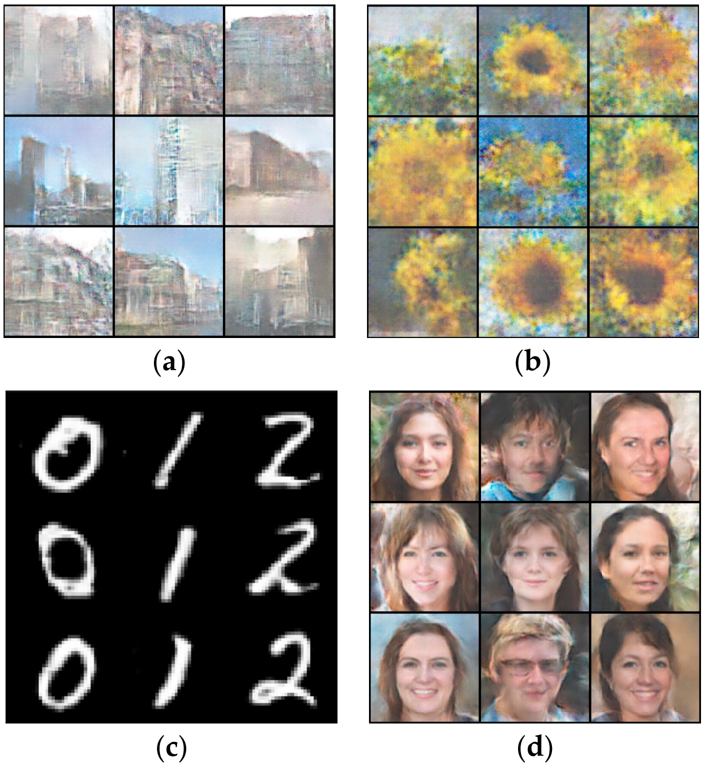

Figure 10. Images containing hidden information: from (**a**) Intel Image Classification, (**b**) Flowers, (**c**) MNIST, and (**d**) Face datasets.

The graph illustrates that when the information embedding dimension is set to 600, the IDGAN model demonstrates remarkable visual performance on the MNIST and Face datasets and performs well on the Flowers dataset. However, the model performs poorly on the Intel Image Classification dataset. This could be due to the large variation in the different architectural styles present in the dataset, which could make it difficult for the model to learn their common features. This indicates that generative steganography's negative impact on image quality can be largely offset in large and complex image datasets. For steganography, a better and more natural quality of the generated image creates higher security.

Table 2 demonstrates that when the input information length is 600 dimensions, the IDGAN restores information completely and with high precision in the MNIST and Face datasets. Although the IDGAN's accuracy rates decrease by 0.7% and 2.3% on the Building and Flowers datasets, respectively, when the information dimension is increased to 720, the Face dataset still ensures a 100% information restoration rate. This suggests that even in large and complex image datasets, the negative impact of generative steganography on the resulting image can be mitigated to some extent. For steganography, the quality of the generated image is critical for higher security.

Table 2. Extraction efficiency.

Data/Dimension	600	720	840
MNIST	100%	100%	97.2%
Intel Image Classification	99.3%	95.4%	92.8%
Flowers	97.7%	93.2%	89.6%
Face	100%	100%	96.2%

4.3.3. FID Quantitative Evaluation

The IDGAN steganography model was redesigned with important modules and loss functions. Thus, it is crucial to evaluate image generation quality and diversity. The FID distance quantification method is used for image evaluation. The aim is to verify whether the GAN steganography algorithm proposed in this paper enhances image quality and diversity while ensuring accurate information recovery. Figure 11 shows the generated image comparison of the GAN and IDGAN models.

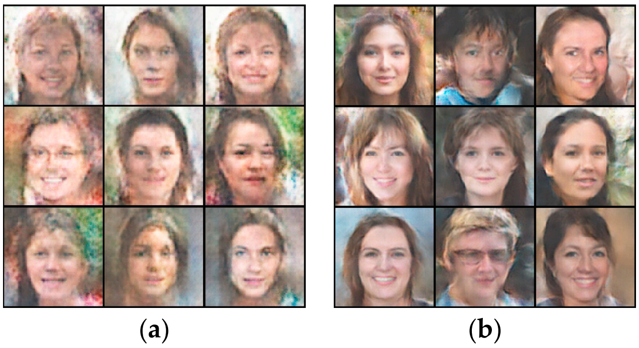

Figure 11. Visual comparison of steganographic images: (**a**) GAN and (**b**) IDGAN.

From the visual impact of the images, it is evident that the IDGAN generated images have fewer high-energy colors in the background, and the facial features are more accurately positioned with less facial distortion, improved hair, and greater facial detail. While the IDGAN's generation capability slightly declined in comparison to a GAN, creating a blurred and shadowy region around the image, it still offers nearly identical diversity in results. Therefore, the IDGAN is a practical solution for image processing requirements.

For a quantitative evaluation of image quality, this chapter uses FID distance quantification. FID is a specialized algorithm designed explicitly for evaluating images generated by GANs. The algorithm measures the degree of similarity between two datasets based on their computer vision features. As real image probability distribution is generally assumed to follow a Gaussian distribution, the FID algorithm applies high-dimensional features as the probability distribution of image data and calculates the image dataset mean and variance. It evaluates the difference between two Gaussian distributions by using the difference between the mean and variance. Higher FID scores indicate an inferior similarity between the two datasets. Therefore, if the target image dataset is highly diverse, comprehensive, and of high quality, a lower FID score indicates stronger generative ability of the model. Table 3 shows the comparison of FID scores of the IDGAN model with other steganography models.

Table 3. Comparison of FID quantitative evaluation.

Model	Zhang's	Zhu's	IDGAN
FID	53.77	48.15	31.26

4.3.4. Steganography Capacity Analysis

From the perspective of steganographic algorithm design, non-embedding steganographic algorithms generally have lower capacity than their embedding counterparts. Therefore, this study compares various non-embedding steganographic algorithms, and their performance is presented in the table. The experiment evaluates the performance of each algorithm using two criteria: steganographic capacity of a single image and embedding rate. In our proposed method, a single image can have a steganographic capacity of 720 bits and an embedding rate of 0.175 bpp. Traditional non-embedding methods, denoted by Method 1 and Method 2 in the table, have steganographic capacities of 6.87×10^{-5} and 0.00169for Zheng and Hu, respectively, indicating lower steganographic efficiency. Zhang et al. employed a non-embedding steganographic approach based on a GAN, with an image size of 64 × 64, a single-image steganographic capacity of 300 bits, and an embedding rate of 0.0732 bpp. On the other hand, Zhu et al. used a non-embedding steganographic method based on generative adversarial networks. Their method has the same image size of 64 × 64, and a maximum embedding rate of 0.264 bpp. However, its corresponding image quality is considerably lower than that of the IDGAN model. Table 4 shows a comparison between the embedding rates of different models.

Table 4. Comparison of embedding rates.

	Bits/Image	**Image Size**	**bpp**
Zheng's [16]	18	512 × 512	0.0000687
Hu's [17]	152	300 × 300	0.00169
Zhang's [25]	300	64 × 64	0.0732
Zhu's [19]	146~1083	64 × 64	0.0356~0.264
IDGAN	720	64 × 64	0.175

The table above compares the performance of different steganography algorithms using two evaluation criteria: single-image steganographic capacity and steganographic embedding rate. The IDGAN model has a single-image steganographic capacity of 720 bits and an embedding rate of 0.17 bpp. Zheng and Hu's model has a low capacity for single-image steganography, with embedding rates of 6.87×10^{-5} and 1.69×10^{-3} for their traditional non-embedding method. Zhang and Zhu's model is a non-embedding steganography method based on generative adversarial networks with the same image size as the IDGAN. Although two of these methods have a higher steganographic capacity, their corresponding image quality is significantly lower than that of the IDGAN. In practice, when the image capacity exceeds 720 bits, the corresponding image quality rapidly deteriorates, despite the ability to recover the complete information by adjusting the training parameters. Therefore, rather than solely focusing on increasing the steganographic capacity, the IDGAN focuses on designing steganography algorithms that ensure desirable image quality.

4.3.5. Security Analysis

Non-embedding steganographic algorithms directly generate steganographic images from secret messages, without using typical cover and secret images in steganalysis models. This enables them to effectively resist steganalysis detection. However, despite having high theoretical security performance, we still need to experimentally analyze their security performance. In our experiment, we will use an improved XuNet as the steganalysis network to detect secret images in different scenarios.

Scenario 1 involves natural and steganographic generated images. The natural image dataset comprises the actual image dataset from the GAN training process, while the generated image dataset is from our algorithm described in this chapter. We will use the steganalysis model to verify the algorithm's security.

Scenario 2 contains images with and without a secret cover, both generated by our algorithm in this chapter. The only difference is whether the noise data are random or structured secret data. The primary objective is to analyze whether an attacker can conduct forensic analysis work after collecting enough historical information when the communication channel is monitored.

Scenario 3 includes non-steganographic and steganographic generated images. The non-steganographic generated image dataset is randomly generated by the SAGAN model, while the steganographic generated image dataset is generated by our algorithm described in this chapter. This scenario aims to verify the resistance of IDGAN-generated secret images against analysis.

Table 5 presents a comparison between the steganographic model and other methods, along with the detection accuracy of the IDGAN. Lower detection accuracy signifies stronger resistance to steganalysis, and as such indicates better security performance.

Table 5. Steganalysis detection accuracy.

Scenario	Detection Rate	
	Other	**IDGAN**
Scenario 1	9.2%	13.5%
Scenario 2	12.4%	13.7%
Scenario 3	67.6%	10.8%

According to the experimental results, the IDGAN demonstrated better performance in image steganography, suggesting that this method holds great potential for practical applications.

5. Threats to Validity

Recently, a considerable number of algorithms and methods have surfaced in the field of image steganography that are based on generative adversarial networks. These have varying impacts on the datasets and tasks they are applied to. It should be noted that the selection of a dataset has a significant effect on a study. While some datasets may yield highly favorable results, others may lead to poor outcomes. Hence, this study aimed to conduct comparative experiments on four datasets in order to provide a more nuanced understanding.

Additionally, despite exhibiting decent performance, the IDGAN model presented in this study has some limitations. Specifically, it may struggle to handle certain types of images, such as complex medical images. This, in turn, can hinder the potential applications of the IDGAN. Furthermore, it is important to consider that the experimental parameters employed can bear a significant influence on the results. Factors such as the dimension and quantity of latent vectors can greatly impact the research outcomes. Lastly, it is noteworthy that visual evaluations conducted in this study were manually analyzed. This could potentially introduce interpretation bias and other subjective judgment issues. Therefore, future research should explore more accurate and objective automated testing methods to mitigate any such distortions.

6. Conclusions

The paper proposes a novel image steganography method based on generative adversarial networks and proves its effectiveness in successfully embedding and extracting information. The attention model related to image steganography is designed and implemented. The loss function of the discriminator is designed using the soft margin loss model, which not only provides the correct training gradient but also leaves redundant space for the steganography process. The extractor's network structure is improved using DenseNet to enhance its convergence ability. The proposed IDGAN is potentially applicable in fields such as digital privacy protection and secure data transmission, possessing high practical value.

Experimental results demonstrate that the IDGAN can successfully hide information in images while maintaining high accuracy and stability in image quality evaluation metrics. In terms of anti-analysis ability, it outperforms traditional carrier-free steganography methods while achieving greater improvement than information-driven generative steganography methods in terms of image quality and information embedding rate. However, increasing the dimension of the hidden information may cause incomplete or damaged information embedding issues. To address this problem, further exploration using deep learning techniques is necessary to improve the accuracy and robustness of information embedding.

Author Contributions: Conceptualization, C.Z. and X.G.; methodology, X.G.; software, X.L. and C.Z.; validation, W.H. and G.Y.; formal analysis, X.G.; investigation, W.H.; resources, L.L.; data curation, L.W.; writing—original draft preparation, X.L. and X.G.; writing—review and editing, X.G. and T.X.; visualization, X.L.; supervision, C.Z.; project administration, C.Z. and X.G.; funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Hebei Province Professional Degree Teaching Case Establishment and Construction Project (Chunying Zhang: No. KCJSZ2022073), the Hebei Postgraduate Course Civic Politics Model Course and Teaching Master Project (Chunying Zhang: No. YKCSZ2021091), the Basic Scientific Research Business Expenses of Hebei Provincial Universities (Liya Wang: No. JST2022001), and the Tangshan Science and Technology Project (Liya Wang: No. 22130225G).

Data Availability Statement: The dataset address used in thins paper is http://yann.lecun.com/exdb/mnist/ (accessed on 1 November 2022), https://www.kaggle.com/datasets/puneet6060/intel-image-classification (accessed on 1 November 2022), https://aistudio.baidu.com/aistudio/datasetdetail/173680/0 (accessed on 3 November 2022).

Acknowledgments: Support by colleagues and the university is acknowledged.

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

References

1. Valandar, M.Y.; Ayubi, P.; Barani, M.J. A new transform domain steganography based on modified logistic chaotic map for color images. *J. Inf. Secur. Appl.* **2017**, *34*, 142–151. [CrossRef]
2. Prasad, S.S.; Hadar, O.; Polian, I. Detection of malicious spatial-domain steganography over noisy channels using convolutional neural networks. *Electron. Imaging* **2020**, *2020*, art00007. [CrossRef]
3. Park, Y.R.; Shin, S.U. Steganographic Method on Spatial Domain Using Modular Characteristic. *J. Korea Inst. Inf. Secur. Cryptol.* **2006**, *16*, 113–119.
4. Pevný, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In Proceedings of the Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, 28–30 June 2010; pp. 161–177.
5. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
6. Rahim, R.; Nadeem, S. End-to-end trained cnn encoder-decoder networks for image steganography. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
7. Volkhonskiy, D.; Nazarov, I.; Burnaev, E. Steganographic generative adversarial networks. *arXiv* **2017**, arXiv:1703.05502.
8. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
9. Shi, H.; Dong, J.; Wang, W.; Qian, Y.; Zhang, X. SSGAN: Secure steganography based on generative adversarial networks. In *Advances in Multimedia Information Processing–PCM 2017, Proceedings of the 18th Pacific-Rim Conference on Multimedia, Harbin, China, 28–29 September 2017*; Revised Selected Papers, Part I 18; Springer: Berlin/Heidelberg, Germany, 2018; pp. 534–544.
10. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
11. Yaojie, W.; Ke, N.; Xiaoyuan, Y. Information hiding scheme based on Generative Adversarial Networks. *J. Comput. Appl.* **2018**, *38*, 6.
12. Tang, W.; Tan, S.; Li, B.; Huang, J. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Process. Lett.* **2017**, *24*, 1547–1551. [CrossRef]
13. 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]
14. Yang, J.; Liu, K.; Kang, X.; Wong, E.K.; Shi, Y.Q. Spatial image steganography based on generative adversarial network. *arXiv* **2018**, arXiv:1804.07939.
15. Li, L.; Fan, M.; Liu, D. AdvSGAN: Adversarial image Steganography with adversarial networks. *Multimed. Tools Appl.* **2021**, *80*, 25539–25555. [CrossRef]
16. Zheng, S.; Wang, L.; Ling, B.; Hu, D. Coverless information hiding based on robust image hashing. In Proceedings of the Intelligent Computing Methodologies: 13th International Conference, ICIC 2017, Liverpool, UK, 7–10 August 2017; pp. 536–547.
17. Hu, D.; Wang, L.; Jiang, W.; Zheng, S.; Li, B. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access* **2018**, *6*, 38303–38314. [CrossRef]
18. Hayes, J.; Danezis, G. Generating steganographic images via adversarial training. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
19. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. 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.
20. Wu, P.; Yang, Y.; Li, X. Stegnet: Mega image steganography capacity with deep convolutional network. *Future Internet* **2018**, *10*, 54. [CrossRef]
21. Duan, X.; Jia, K.; Li, B.; Guo, D.; Zhang, E.; Qin, C. Reversible image steganography scheme based on a U-Net structure. *IEEE Access* **2019**, *7*, 9314–9323. [CrossRef]

22. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
23. Fu, Z.; Wang, F.; Cheng, X. The secure steganography for hiding images via GAN. *EURASIP J. Image Video Process.* **2020**, *2020*, 46. [CrossRef]
24. Mo, L.; Zhu, L.; Ma, J.; Wang, D.; Wang, H. MDRSteg: Large-capacity image steganography based on multi-scale dilated ResNet and combined chi-square distance loss. *J. Electron. Imaging* **2021**, *30*, 013018. [CrossRef]
25. Lu, S.P.; Wang, R.; Zhong, T.; Rosin, P.L. Large-capacity image steganography based on invertible neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10816–10825.
26. Zhang, R.; Dong, S.; Liu, J. Invisible steganography via generative adversarial networks. *Multimed. Tools Appl.* **2019**, *78*, 8559–8575. [CrossRef]
27. 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]
28. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.