Attention Based Data Hiding with Generative Adversarial Networks

Chong Yu[1]
[1]NVIDIA Corporation
No.5709 Shenjiang Road, No.26 Qiuyue Road
Shanghai, China 201210
dxxzdxxz@126.com, chongy@nvidia.com

Abstract

Recently, the generative adversarial network is the hotspot in research and industrial areas. Its application on data generation is the most common usage. In this paper, we propose the novel end-to-end framework to extend its application to data hiding area. The discriminative model simulates the detection process, which can help us understand the sensitivity of the cover image to semantic changes. The generative model is to generate the target image which is aligned with the original cover image. An attention model is introduced to generate the attention mask. This mask can help to generate a better target image without perturbation of the spotlight. The introduction of cycle discriminative model and inconsistent loss can help to enhance the quality of the generated target image in the iterative training process. The training dataset is mixed with intact images and attacked images. The mix training process can further improve robustness. Through the qualitative, quantitative experiments and analysis, this novel framework shows compelling performance and advantages over the current state-of-the-art methods in data hiding applications.

In the information era, a large amount of data is shared through the Internet at every moment. The barrier of accessing data is much lower, which leads to more potential violations of data security, privacy, copyright, etc. Data hiding technologies are widely used to solve the aforementioned problems in secret communication (Holub, Fridrich, and Denemark 2014), digital watermarking (Yu 2016), cryptography (El Hossaini et al. 2016), etc. While some attack methods (Shi et al. 2017) pose the threat to data hiding technologies.

In Figure 1, can you easily differentiate between Van Gogh's paintings in (a) and (b)? Or Monet's paintings in (e) and (f)? Actually, the images in (a) and (e) are the original version of drawing masters' works. Images in (b) and (f) are data hiding version generated by our method. The hidden images are emblems of painters' nations: Netherland and France. The embedded info is kept imperceptible to ensure there is no influence on the audience to appreciate paintings from the fidelity aspect.

In this paper, we propose an **A**ttention **B**ased **D**ata **H**iding framework with Generative Adversarial Networks (GAN),

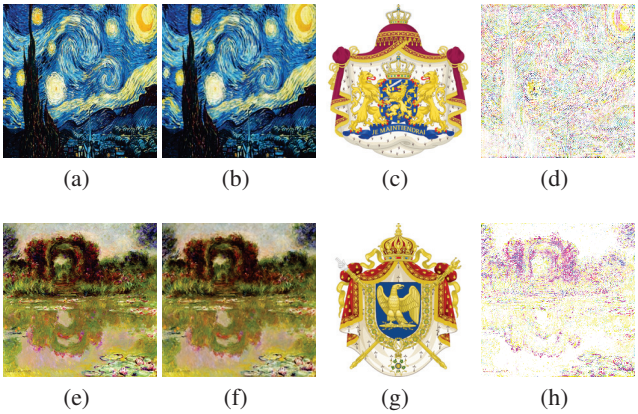

Figure 1: Illustration of data hiding performance on the world-renowned art paintings. (a,e) Original version of *The Starry Night* painted by *Van Gogh* and *The Rose Arches* by *Monet*. (b,f) Data hiding version of *The Starry Night* and *The Rose Arches*. (c,g) *Emblem of Netherland* and *France* as the hidden images. (d,h) Residual difference between original and data hiding versions. (We multiply the residuals with constant value 10 to emphasize the difference.)

and use **ABDH** as the acronym to represent the method in this paper. The ultimate goal of data hiding is making hidden data imperceptible to the detector. So adversarial relation always exists in data hiding applications. **ABDH** combines the evaluation metrics of secure data hiding with the advantages in the latest GAN principle, and integrates the counterparts into a single framework. The attention mechanism is introduced to further guide **ABDH** to find the inconspicuous areas of cover images, which is suitable for hiding secret data. With the fine-tuning adversarial training process, **ABDH** can iteratively learn to robustly hide data in cover images or videos in an end-to-end manner.

The main contribution of our work includes:

- **ABDH** is general for data hiding applications. It can apply to steganography and watermarking applications.
- **ABDH** simulates the hidden data detection process with

the discriminative model. It helps the generative model on understanding the sensitivity of cover images.

- The introduction of attention mechanism helps the generative model to aware of spotlights and inconspicuous areas of cover images.
- **ABDH** learns to resist various attacks (noise, crop, compression, etc.) in an end-to-end manner.

Related Work

Data Hiding

Data hiding (Bansal et al. 2016) is referred as embedding additional data in a cover medium such as image, video, audio, and file. Data hiding technology is widely used in secret information transmission (Shi et al. 2017), watermarking (Yu 2016), copyright certification (Mun et al. 2017), forgery detection (Wolfgang and Delp 1996) applications. Among them, steganography and watermarking are two hotspots in research and industrial areas. They are the focused application areas of **ABDH**.

Steganography

Steganography literally means "covered writing" and is usually interpreted to hide information in other information. Steganography methods can be categorized into three types.

Least Significant Bit Steganography The main strength of this category is that algorithms are theoretically simple and have low computational complexities. Secret information is embedded into the cover image with the operations like shifting or replacing of pixels. In typical Least Significant Bit (LSB) algorithm, pixel values of the cover image and secret messages are represented by binary form. Stego image generation process is implemented by replacing the least significant bits of cover image with the most significant bits of secret information. In (Das, Samaddar, and Keserwani 2018), authors proposed to generate an LSB based hash function for the image authentication process, which can provide good imperceptibility between the original image and stego image with hash bits.

Content Adaptive Steganography Algorithms in this category design the hand-crafted distortion functions which are used for selecting the embedding localization of the image. Wavelet Obtained Weights (WOW) (Holub and Fridrich 2012) embeds information into the cover image according to the textural complexity of regions. Highly Undetectable Steganography (HUGO) (Pevnỳ, Filler, and Bas 2010) defines a distortion function domain by assigning costs to pixels based on the effect of embedding some information within a pixel. It uses a weighted norm function to represent the feature space. S-UNIWARD (Holub, Fridrich, and Denemark 2014) proposes a universal distortion function that is independent of the embedded domain. They are all devoted to minimize distortion functions, to embed the secret into the noisy area or complex textures, and to avoid the smooth regions of the cover images.

Deep Learning Based Steganography As deep learning has a brilliant capability in image processing, researchers also attempt to utilize it in steganography. Paper (Volkhonskiy et al. 2017) introduces a new model for generating more steganalysis-secure cover images based on deep convolutional GAN. Based on Wasserstein GAN (Arjovsky, Chintala, and Bottou 2017), paper (Shi et al. 2017) proposes algorithm which is efficient to generate cover images with higher visual quality. Paper (Dong, Zhang, and Liu 2018) proposes a steganography model that can conceal a gray secret image into a color cover image with the same size, and generate stego image which seems quite similar to cover image in semantics and color. Paper (Zhu et al. 2018) introduces HiDDeN, which is an end-to-end trainable framework that works for both steganography and watermarking applications.

Watermarking

Watermarking is defined as the process of embedding a message, called "watermark" into images, videos, and audio files. There are two main differences between steganography and watermarking. Firstly, the information embedded by watermarking is always associated with the digital object to be protected, while steganography just hides any information that needs to be imperceptible. Secondly, "robustness and safety" criteria are different. Steganography mainly concerns the detection of the hidden message, while watermarking concerns potential removal by a pirate.

For watermarking, robustness is a vital feature. It means the ability of embedded watermark to resist common image processing operations. According to robustness character, watermarking techniques can be further divided into two categories: fragile and robust watermarking. The aim of fragile watermarking is to identify and detect every possible tampering in the watermarked digital objects. So the embedded fragile watermark is very sensitive to the modification. On the opposite, the robust watermark should survive against a multiplicity of attacks such as cropping, scaling, filtering, additive noise, and JPEG compression. Digital watermarking methods can be categorized into three types.

Spatial Domain Watermarking In the spatial domain, watermarking is done in the pixel domain, with advantages such as low complexity, low cost, and low delay. Paper (Banitalebi, Nader-Esfahani, and Avanaki 2018) proposes a robust LSB watermarking method that utilizes structural similarity in the embedding and extraction rules.

Spectral Domain Watermarking In the spectral domain, watermarking is achieved by various transform domain technologies. Discrete Cosine Transform (DCT) is a favored transform function. In theory, pixel bit values are firstly transformed using DCT, then added to the cover image's DCT coefficients. Because the embedding process is in the spectral domain, the watermark is more imperceptible in the spatial domain. Paper (Parah et al. 2016) exploits the correlation between DCT coefficients of adjacent blocks. The inter-block coefficient difference is the key to decide the amount of watermark embedded. Discrete Wavelet Transform (DWT) is another promising method. In (Lu et al.

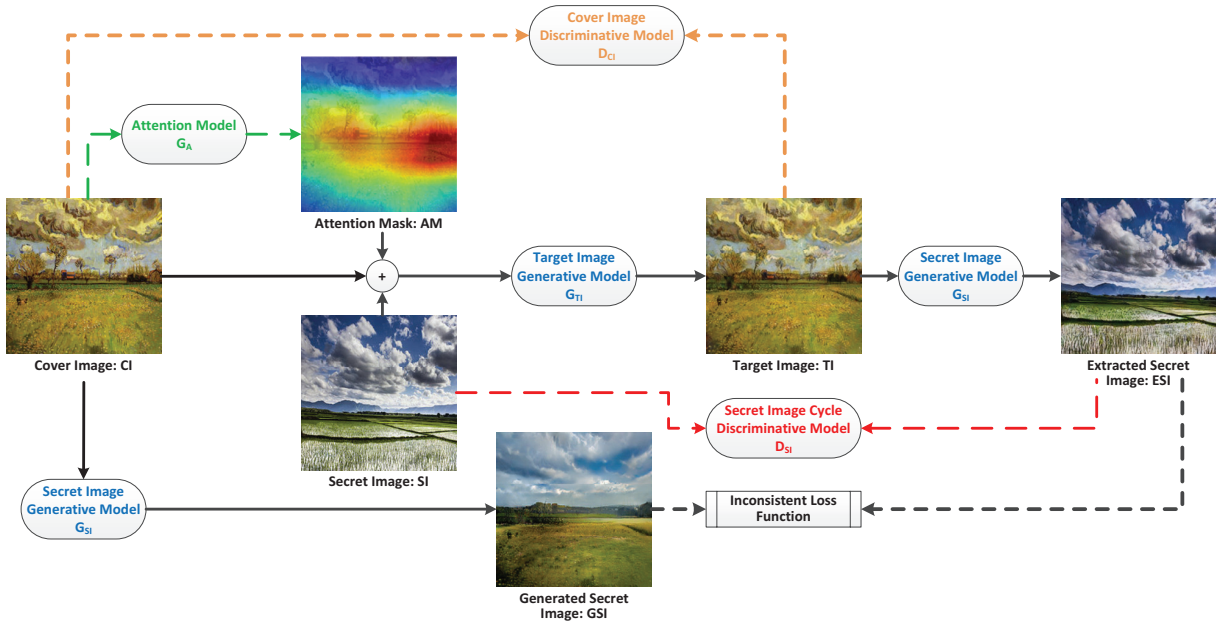

Figure 2: Framework and workflow chart of **ABDH**.

2015), DWT is applied to decompose cover image and watermark into the low-frequency and high-frequency sub-band components. By embedding the watermark in the low-frequency sub-band, the watermarked image can have better imperceptibility and higher robustness.

Neural Network Based Watermarking Paper (Mun et al. 2017) introduces a blind watermarking based on the neural network that can detect a 1-bit message from an image sub-block. The loop of the learning process consists of watermark embedding, attack simulation, and weight update stages. In (Ferdowsi and Saad 2017), a deep learning method based on long short-term memory structure is proposed for the dynamic watermarking. It enables the communication device to extract stochastic features from its generated signal and dynamically watermark these features into the signal.

Attention Based Data Hiding

Principle

For data hiding applications, the common aim is to embed the secret message into the cover message. So we consider essential evaluation metrics when designing **ABDH**[1].

- The secret message should remain imperceptible until it is extracted by specific authorized receiver.
- The target image should be robust and intact to resist tampering and attacks.

For the first evaluation metric, if an eavesdropper wants to check whether the image he obtained from public media contains a hidden secret message. So he needs to discriminate the original cover image and received target image. If these two images are perceptibly same, then the eavesdropper can hardly differentiate the target image from the cover image. Then the threat of finding the secret message is lower. For the purpose of data hiding, we can accumulate the visual and statistic differences between cover and target images. If the difference for each evaluation metric is small enough, we can regard the target image as a high-quality production.

For the second evaluation metric, if the eavesdropper wants to destroy secret communication. So he makes intentional changes to the target image, like rotate, clip, add noises and make compression. Because he assumes that even the image he obtained contains a secret message, these intentional changes will make the secret message extraction method disabled. If the data hiding method is robust and secure, the intentional changes are in vain.

GAN (Goodfellow et al. 2014) consists of the generative model and the discriminative model. The purpose of the generative model is to generate new samples that are very similar to the real samples and attempts to confuse the discriminator. While the purpose of the discriminative model is to classify samples synthesized by the generative model and the real ones. The discriminative model will also estimate the probability that a specific sample comes from the generative model rather than the real ones. When the whole GAN model achieves Nash Equilibrium, i.e., the generative model can generate the samples which exactly align with the character and distribution of real samples. And at the same time, the discriminative model returns the classification probability 0.5 for each pair of generated and real samples. Then this GAN model is well-trained and converged.

ABDH method combines the purpose of data hiding with the GAN principle. It consists of a target generative and a

[1]In this paper, we use the **cover message** to represent the host of hiding data. The **secret message** refers to the hidden data. The **target message** refers to the production that secret message is already hidden into the cover message.

hidden data discriminative model. The purpose of the target generative model is to generate the target image which is very similar to the cover image and attempts to confuse the discriminative model. While the purpose of the discriminative model is to distinguish the generated target image from the cover image. When **ABDH** achieves Nash Equilibrium, i.e., the generative model can keep the secret message imperceptible in the target image. And at the same time, the hidden data discriminative model cannot detect the existence of a secret message for each pair of target and cover images. This also aligns with the imperceptible and robust evaluation criterions of data hiding. In conclusion, designing a data hiding algorithm is equal to make the **ABDH** model converged.

Algorithm

In **ABDH**, there are two generative models and two discriminative models. For a general data hiding framework, it should contain secret message embedding and extraction processes. So it needs to learn the bijective mapping relationship between two image collections. For **ABDH**, one image collection contains the original cover images, the other collection contains the secret images for hiding. Framework of **ABDH** is shown in Figure. 2.

In the left part of Figure 2, the original **C**over **I**mage (*CI*) goes through the attention model $\boldsymbol{G}_A$, to produce the **A**ttention **M**ask (*AM*). The attention model used in **ABDH** is the feature extraction backbone of ResNet50[2] (He et al. 2016). The intuition comes from recent deep network visualization research (Simonyan, Vedaldi, and Zisserman 2013) (Zhou et al. 2016). These works find the activation map can build a generic localizable representation that exposes the implicit attention of deep neural networks on images. This is the imitation of the human attention mechanism. Because the data hiding task is trying to "confuse" the visual effect between the cover image and target image, so we introduce this attention mask to help **ABDH** to learn this feature explicitly. The attention mask generation process can be expressed as follows.

$$AM = \boldsymbol{G}_A(CI) \tag{1}$$

Each value of *AM* represents the "attention sensitiveness" of each pixel in *CI*. The value is regularized to the range from 0 to 1. If the value is closer to 1, it means the change of the corresponding pixel will lead to obvious differences, and easily cause the attention of visual detection. After *AM* is generated, the original *CI*, *AM* and original **S**ecret **I**mage (*SI*) go through the target image generative model $\boldsymbol{G}_{TI}$, to produce the **T**arget **I**mage (*TI*). This is the secret embedding and target image generation process, which can be expressed as follows.

$$TI = \boldsymbol{G}_{TI}(CI, SI, AM) \tag{2}$$

In the right part of Figure 2, the target image *TI* goes through the secret image generative model $\boldsymbol{G}_{SI}$, to get the **E**xtracted **S**ecret **I**mage (*ESI*). This is the secret image extraction process, which can be expressed as follows.

$$ESI = \boldsymbol{G}_{SI}(TI) \tag{3}$$

[2]ResNet50 structure and pre-trained model are from the repository: https://github.com/pytorch/vision.

The cover image discriminative model $\boldsymbol{D}_{CI}$ ensures that the distribution of *CI* is indistinguishable from that of *TI* using an adversarial loss. This is the guarantee of the imperceptible evaluation criterion in data hiding.

To refine the secret extraction process, we introduce the secret image cycle discriminative model $\boldsymbol{D}_{SI}$. Because the generative model is learned to transform from a source image domain to a target image domain. Take $\boldsymbol{G}_{SI}$ as an example, the learned mapping relation is highly underconstrained, and cannot ensure the extracted secret image *ESI* is indistinguishable from the original secret image *SI* (Zhu et al. 2017). So we couple this mapping relation with its inverse mapping $\boldsymbol{G}_{TI}$, and introduce a cycle adversarial loss:

$$\boldsymbol{D}_{SI}(SI, ESI) \rightarrow 0 \tag{4}$$

That is equal to

$$ESI = \boldsymbol{G}_{SI}(TI) = \boldsymbol{G}_{SI}(\boldsymbol{G}_{TI}(CI, SI, AM)) \approx SI \tag{5}$$

It ensures the distribution of *ESI* is indistinguishable from that of *SI* using cycle adversarial loss $\boldsymbol{D}_{SI}$. This is the guarantee of the secure and robust extraction criterion in data hiding.

To guarantee the uniqueness property, we introduce the extra inconsistent loss. This loss ensures the secret image can only be extracted from the target image *TI*. If we apply the secret image extraction process to the original cover image *CI*, the **G**enerated fake **S**ecret **I**mage (*GSI*) should be totally different from the original secret image *SI*. The inconsistent loss can be expressed as follows:

$$\max_{\boldsymbol{G}_{SI}}(GSI, ESI) = \max_{\boldsymbol{G}_{SI}} |\boldsymbol{G}_{SI}(CI) - \boldsymbol{G}_{SI}(TI)| \tag{6}$$

The detailed algorithm workflow of **ABDH** is summarized as Algorithm 1.

Loss Function

The overall loss function of **ABDH** consists of three parts: the adversarial loss $L_{GAN}(\boldsymbol{G}_{TI}, \boldsymbol{D}_{CI})$, the cycle adversarial loss $L_{GAN}(\boldsymbol{G}_{SI}, \boldsymbol{D}_{SI})$ and the inconsistent loss L_{IC}. So the loss function is written as follows:

$$\begin{aligned} L_{Overall} = L_{GAN}(\boldsymbol{G}_{TI}, \boldsymbol{D}_{CI}) + L_{GAN}(\boldsymbol{G}_{SI}, \boldsymbol{D}_{SI}) \\ +\lambda L_{IC}[\boldsymbol{G}_{SI}(CI), \boldsymbol{G}_{SI}(TI)], \end{aligned} \tag{7}$$

where λ is the parameter to adjust the percentages between adversarial loss and inconsistent loss. The inconsistent loss needs to change to the minimization format as follows.

$$\min_{\boldsymbol{G}_{SI}} \frac{1}{|\boldsymbol{G}_{SI}(CI) - \boldsymbol{G}_{SI}(TI)|} \tag{8}$$

In the **ABDH** framework, the quality of the generated target image *TI* and the extracted secret image *ESI* are judged by the difference from the original cover image *CI* and original secret image *SI*, respectively. In this paper, two quantitative image effect indicators are applied to measure the differences (Yu 2016). Peak Signal to Noise Ratio (PSNR) indicator is applied to assess the effect difference in the gray-level fidelity aspect. Structural Similarity (SSIM) (Wang et

Table 1: Evaluation metrics of generated target images *TI*.

Metrics/Images	Lena	Airplane	Baboon	Fruits	Peppers
PSNR	33.0170	33.0065	29.1163	33.9085	30.5124
SSIM	0.9390	0.9589	0.9335	0.9510	0.9034

Table 2: Evaluation metrics of extracted secret images *ESI*.

Metrics/Images	from Lena	from Airplane	from Baboon	from Fruits	from Peppers
PSNR	30.6247	30.9730	31.1053	30.2076	30.4095
SSIM	0.9530	0.9563	0.9573	0.9488	0.9508

Algorithm 1 ABDH Algorithm

Input: Training set of cover images and secret images
Parameter: Loss adjustment λ, Overall loss threshold δ
Output: Target image generative model $\boldsymbol{G}_{TI}$, Secret image generative model $\boldsymbol{G}_{SI}$

Initialize generative and discriminative models with random value.
Initialize attention model with pre-trained ResNet-50.
while $L_{Overall} > \delta$ **do**
  if Target image generation sub-process **then**
    - Attention mask generation: *AM* = $\boldsymbol{G}_A$(*CI*)
    - Secret embedding: *TI* = $\boldsymbol{G}_{TI}$(*CI*, *SI*, *AM*)
    - Minimize difference between *CI* and *TI*, e.g., Adversarial loss: $L_{GAN}(\boldsymbol{G}_{TI}, \boldsymbol{D}_{CI})$
  end if
  if Secret extraction sub-process **then**
    - Extract from *TI*: *ESI* = $\boldsymbol{G}_{SI}$(*TI*)
    - Minimize difference between *SI* and *ESI*, e.g., Cycle adversarial loss: $L_{GAN}(\boldsymbol{G}_{SI}, \boldsymbol{D}_{SI})$
    - Extract from *CI*: *GSI* = $\boldsymbol{G}_{SI}$(*CI*)
    - Maximize difference between *ESI* and *GSI*, e.g., Inconsistent loss: L_{IC}(*ESI*, *GSI*)
  end if
  - Minimize: $L_{Overall} = L_{GAN}(\boldsymbol{G}_{TI}, \boldsymbol{D}_{CI}) + L_{GAN}(\boldsymbol{G}_{SI}, \boldsymbol{D}_{SI}) + \lambda L_{IC}(ESI, GSI)$
  - Update $\boldsymbol{G}_{TI}, \boldsymbol{G}_{SI}, \boldsymbol{D}_{CI}, \boldsymbol{D}_{SI}$
  - Learning rate decay
end while
return Converged generators: $\boldsymbol{G}_{TI}, \boldsymbol{G}_{SI}$

al. 2004) indicator which is an image quality assessment indicator based on the human vision system is applied to assess the effect difference in the structure-level fidelity aspect. The definitions of these two evaluation indicators are as follows.

$$PSNR(\boldsymbol{x},\boldsymbol{y}) = 10\log_{10}\left(\frac{(MAX_I)^2}{MSE(\boldsymbol{x},\boldsymbol{y})}\right), \quad (9)$$

where MAX_I is the maximum possible pixel value of images $\boldsymbol{x}$ and $\boldsymbol{y}$. $MSE(\boldsymbol{x},\boldsymbol{y})$ represents the Mean Squared Error (MSE) between images $\boldsymbol{x}$ and $\boldsymbol{y}$.

$$SSIM(\boldsymbol{x},\boldsymbol{y}) = \frac{(2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{y}} + C_1)(2\sigma_{\boldsymbol{xy}} + C_2)}{(\mu_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{y}}^2 + C_1)(\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{y}}^2 + C_2)}, \quad (10)$$

where $\mu_{\boldsymbol{x}}$ and $\mu_{\boldsymbol{y}}$ represent the average grey values of images. Symbol $\sigma_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{y}}$ represent the variances of images. Symbol $\sigma_{\boldsymbol{xy}}$ represents covariance between images. C_1 and C_2 are two constants which are used to prevent unstable results when either $\mu_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{y}}^2$ or $\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{y}}^2$ is very close to 0.

Network Structure

The network structure of target image generative model $\boldsymbol{G}_{TI}$ includes a convolution layer (kernel size = 7, stride = 0, pad = 0), two convolution layers (k = 3, s = 2, p = 1), nine residual blocks (He et al. 2016), and two deconvolution layers (k = 3, s = 2, p = 1, outside pad = 1), and a convolution layer (k = 7, s = 0, p = 0). Each convolution and deconvolution layer follows with an instance normalization layer and a ReLU layer. The structure of secret image generative model $\boldsymbol{G}_{SI}$ is identical with $\boldsymbol{G}_{TI}$.

The network structure of cover image discriminative model $\boldsymbol{D}_{CI}$ is similar with PatchGAN model (Isola et al. 2017). Each time, it operates an image patch with 70×70 size, and classifies whether this patch is real or fake. The model will run across the whole image, and average all results in the 70×70 overlapping patches to provide the ensemble output. The architecture of such a patch-level discriminative model requires fewer parameters and runs faster than a full-image discriminator (Yi et al. 2017). Moreover, it has no constraints over the size of the input image. $\boldsymbol{D}_{CI}$ contains a convolution layer (k = 4, s = 2, p = 1) follows with a leaky ReLU layer, three convolution layers (k = 4, s = 2, p = 1) follows with an instance normalization layer and a leaky ReLU layer, a convolution layer (k = 4, s = 1, p = 1) follows with an instance normalization layer and a leaky ReLU layer, a convolution layer (k = 4, s = 1, p = 1) follows with a sigmoid layer to output a scalar output between [0, 1]. The structure of secret image cycle discriminative model $\boldsymbol{D}_{SI}$ is identical with $\boldsymbol{D}_{CI}$.

Moreover, to improve the convergence performance, we use Adam optimizer (Kinga and Adam 2015) instead of stochastic gradient descent (SGD) optimizer. It is computationally efficient and has little memory requirements. The hyper-parameters of Adam optimizer are: β_1=0.5, β_2=0.999. The base learning rate is 0.0002.

Table 3: *PSNR* metric for extracted secret images.

Images/Algorithms	LSB-TLH	WOW	HUGO	S-UNIWARD	DCT-ICD	DWT-DCT	ISGAN	SSGAN	HiDDeN	DS	**ABDH**
Gaussian noise	24.8580	24.7123	27.1831	26.5263	26.3086	26.0104	22.4100	23.6919	25.8199	25.7353	**33.2285**
Possion noise	27.1579	27.1803	27.1612	27.1810	27.0854	27.0754	27.1757	27.1746	27.1701	25.8348	**27.1987**
Salt and Pepper noise	19.2052	21.2156	24.6026	21.3960	24.2090	24.1357	19.8900	20.5929	21.6048	25.5675	**33.4822**
Salt noise	31.5330	32.0738	35.3124	34.3301	34.2935	34.3127	31.0912	31.9889	33.3123	26.6439	**45.0600**
Pepper noise	17.0777	17.3501	18.1149	19.2487	21.1004	21.1223	16.1582	17.2083	17.9478	24.5692	**29.3407**
Speckle noise	23.5170	24.6379	25.5571	27.6822	27.2157	29.7533	21.9666	22.4623	25.3485	28.0589	**46.4994**
JPEG Compression	30.3018	32.1750	32.4679	31.3277	31.3444	32.4001	30.9556	31.2820	31.5681	31.4456	**32.9739**
Low-pass filter	27.9778	33.8242	32.7279	30.5234	32.9714	32.5547	28.8135	29.6587	31.2633	30.4309	**37.5805**
High-pass filter	19.7074	20.2493	22.0935	20.4663	20.0646	20.0865	19.6455	19.7446	20.6291	20.4875	**22.8699**
Median filter	21.9184	22.5196	25.4038	23.6964	25.6347	25.5186	21.0874	22.0219	23.3846	23.0524	**28.5659**
Random crop 10%	28.0956	28.1163	28.1097	28.1035	**28.6845**	28.5479	28.0996	28.1182	28.1063	28.1078	28.1181
Random crop 20%	20.6125	22.3013	22.3739	22.1683	22.5010	22.3764	20.2038	21.0407	21.8640	21.4467	**23.3684**
Random shift 10%	26.6862	30.4574	31.0763	29.7196	30.9949	30.6371	27.9319	28.8819	29.4849	29.4024	**31.6285**
Random shift 20%	21.8021	22.4331	22.5555	24.2972	23.2096	23.1073	21.7754	21.8409	22.7720	22.6172	**25.0980**

Experimental Results

To train **ABDH**, we apply the COCO dataset (Lin et al. 2014). We randomly divide COCO with the 8:2 ratio to generate the separate training and validation datasets. In the training dataset, we randomly choose 50% as cover images, and 50% as secret images. We crop original images to 512×512. For the original images with a smaller size, we resize them to 550×550, then crop to 512×512. To improve the robustness against attacks, we also generate an attacked training dataset with the same number of the original training dataset. From the original training dataset, we randomly select 10% samples each to add multiplicative noise, salt and pepper noise, gaussian white noise, Poisson noise, low-pass filter, high-pass filter, median filter, random crop, random shift, and 10% to do JPEG compression. We add the JPEG compression to simulate the coding and decoding processes in the real secure information transmission system. We need to ensure **ABDH** can work well against coding and decoding algorithms. Parameters of each noise (like mean and variance of random distribution, JPEG compression quality, etc.) are randomly generated for each image. The testing dataset is generated by combining Set5 (Bevilacqua et al. 2012) and Set14 (Zeyde, Elad, and Protter 2010) datasets. We use PyTorch as the framework and train **ABDH** with 150 epochs. The loss adjustment parameter λ is set as 0.6.

In experiments, we want to investigate these issues:

- **ABDH** performance experiments.
- Quantitative comparison with state-of-the-art methods.
- Ablation experiments.
- Influence of embedding secret info amount.

Data Hiding Performance Experiments

We adopt the benchmark images from Set5 and Set14 datasets as the cover images *CI* shown in **Row 1** of Figure 3 to test the data hiding performance of **ABDH**, include the target image generation and secret extraction processes. The embedded secret image is a poster image. We choose this image because it is never seen in training and validation datasets, and it has enough complexity with characters in it. The generated attention masks are shown in **Row 2** of Figure 3. We use different colors to represent different attention-sensitive degrees. The color which is more similar to red represents more attention-sensitive areas, while color which is more similar to blue represents less attention-sensitive areas.

PSNR and *SSIM* metrics for generated target images *TI* versus cover images *CI* are shown in Table 1. (*TI* and *CI* are used as image $\boldsymbol{x}$ and $\boldsymbol{y}$ for metrics calculation) The results shown in Figure 3 and Table 1 can prove the high quality and difference imperceptibility of *TI* in qualitative and quantitative aspects.

Let's have a further analysis of the obtained results. If we magnify Figure 3 to see the residual differences, we can find they are mainly on the marginal and textural parts of objects. For example, the hat of Lena, the edges of F16 plane, the skin and whiskers of baboon, the profile of fruits and peppers, etc. It means **ABDH** tends to hide the secret info into marginal parts of the object in original cover images. In information theory, textures and edges represent the high-frequency parts of the image, while smooth regions represent the low-frequency parts of the image. If we change the low-frequency parts, it is easy to be detected. So many state-of-the-art algorithms transform the cover image from spatial domain to frequency domain. Change the tiny part in high-frequency parts, and transform it back to the spatial domain. Moreover, when we discuss the state-of-the-art content adaptive steganography and spectral domain watermarking algorithms, we find the ultimate goal is trying to embed the secret image into the parts with complex edges and textures, and avoiding the smooth regions of the cover images. The behavior of **ABDH** is very similar to the state-of-the-art steganography and watermarking algorithms. But the state-of-the-art algorithms need to design a hand-crafted distortion function to achieve the goal, while **ABDH** learns from the discriminative network which simulates the behaviors of the detector. From the learning process, the generative network in **ABDH** finds detector are very sensitive to the low-frequency parts, and not so sensitive to the high-frequency parts. So the target images generated by **ABDH** mainly hide their secret info into marginal and textural parts to ensure the best imperceptibility.

PSNR and *SSIM* metrics for extracted *ESI* versus original secret images *SI* are shown in Table 2. (*ESI* and *SI* are used as image $\boldsymbol{x}$ and $\boldsymbol{y}$ for metrics calculation) The results shown

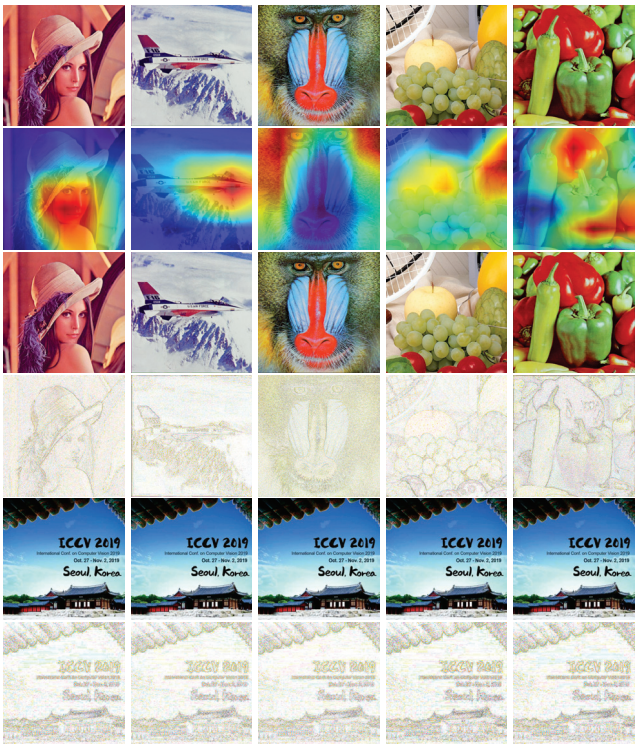

Figure 3: Data hiding performance of **ABDH**. **Row 1:** Original cover images from Set-5/14 datasets. **Row 2:** Generated attention mask. **Row 3:** Generated target images. **Row 4:** Residual difference between cover and target images. **Row 5:** Extracted secret images. **Row 6:** Residual difference between original and extracted secret images. (Because the differences are inconspicuous, so we multiply the residuals with constant value 10 to emphasize in Row 4 and 6.)

in Figure 3 and Table. 2 can prove the high secret recovery quality of *ESI* in qualitative and quantitative aspects.

Quantitative Comparative Experiments

In this experiment, we evaluate the robustness when recovering the secret image from the target image. For LSB steganography and spatial domain watermarking, we choose *LSB-TLH* (Das, Samaddar, and Keserwani 2018). For content adaptive steganography, we choose *WOW* (Holub and

Figure 4: Contribution of key features. **Column 1:** Original cover image. **Column 2:** Target image generated by full **ABDH**. **Column 3-5:** Target image generated by **ABDH** without cycle discriminative model, without inconsistent loss, and without attention model.

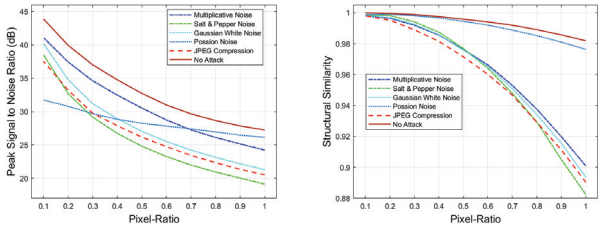

Figure 5: *PSNR* and *SSIM* metrics for *TI* versus *CI* with different pixel-ratio and attacks.

Fridrich 2012), *HUGO* (Pevnỳ, Filler, and Bas 2010) and *S-UNIWARD* (Holub, Fridrich, and Denemark 2014). For spectral domain watermarking methods, we choose *DCT-ICD* (Parah et al. 2016) and *DWT-DCT* (Lu et al. 2015). For deep learning based steganography and watermarking, we choose *ISGAN* (Dong, Zhang, and Liu 2018), *SSGAN* (Shi et al. 2017), *HiDDeN* (Zhu et al. 2018) and *DS* (Baluja 2017) for comparation. Benchmark images in Set-5 are used as the secret images. Benchmark images in Set-14 are used as original cover images. *PSNR* and *SSIM* metrics for recovered secret images are shown in Table 3 $\sim$ 4. They are the average values in the Set-5/14 datasets. According to these metrics, the performance and robustness of **ABDH** outperforms all other state-of-the-art watermarking and steganography algorithms in quantitative aspect.

To have a further analysis of the obtained results, we can find the performance of deep learning based methods is not as good as expected. For *SSGAN*, it is focused on generating the new cover images which are steganalysis-secure. But in our experiments, the cover images are fixed. For *HiDDeN*, bits per pixel that can be hidden is very low, so it is not suitable for whole image embedding. For *ISGAN*, its inherent limitation is extracted secret image will be lossy when the target image is attacked by noise. For *DS*, it does not consider noise attack when training the steganalysis network. So the performance of deep learning steganography methods is just at the same level of LSB steganography methods and is worse than content adaptive steganography methods.

Ablation Experiments

In this experiment, we want to check the contribution of each component in **ABDH** to the final data hiding effect. Then we can have a deep insight into why **ABDH** can outperform state-of-the-art methods. We include three key features which have the potential big contribution.

- The introduction of cycle discriminative model.
- The introduction of extra inconsistent loss.
- The introduction of attention model.

In the experiments, three extra **ABDH** models are trained on COCO training dataset. Each model lacks one of aforementioned feature. We adopt the benchmark images from Set5 and Set14 datasets as the secret images and cover images. We calculate the *PSNR* and *SSIM* metrics for generated target images *TI* versus cover images *CI*, which are shown in Table 5. To illustrate the influences of these three features

Table 4: *SSIM* metric for extracted secret images.

Images/Algorithms	LSB-TLH	WOW	HUGO	S-UNIWARD	DCT-ICD	DWT-DCT	ISGAN	SSGAN	HiDDeN	DS	**ABDH**
Gaussian noise	0.6662	0.6579	0.7674	0.7403	0.7542	0.8143	0.5403	0.6079	0.7080	0.7219	**0.9278**
Possion noise	0.7620	0.7631	0.7626	0.7628	0.7619	0.7623	0.7626	0.7628	0.7626	0.7628	**0.7637**
Salt and Pepper noise	0.6349	0.7492	0.8744	0.7563	0.8221	0.8422	0.6774	0.7168	0.7537	0.7933	**0.9839**
Salt noise	0.9944	0.9950	0.9974	0.9968	0.9923	0.9985	0.9939	0.9949	0.9959	0.9964	**0.9997**
Pepper noise	0.4860	0.5048	0.5608	0.6364	0.8441	0.8898	0.4175	0.4936	0.5470	0.5673	**0.9554**
Speckle noise	0.5847	0.6324	0.6828	0.7646	0.7351	0.7521	0.4978	0.5310	0.6661	0.6933	**0.9976**
JPEG Compression	0.9876	0.9910	0.9909	0.9899	0.9887	0.9873	0.9898	0.9898	0.9898	0.9876	**0.9914**
Low-pass filter	0.8955	0.9635	0.9542	0.9304	0.9623	**0.9845**	0.9062	0.9206	0.9300	0.9386	0.9836
High-pass filter	0.7147	0.7384	0.8096	0.7476	0.7314	0.7323	0.7127	0.7156	0.7446	0.7521	**0.8352**
Median filter	0.6548	0.6802	0.7885	0.7271	0.8298	0.8422	0.6182	0.6589	0.6938	0.7035	**0.8772**
Random crop 10%	0.8876	0.8878	0.8878	0.8877	**0.8908**	0.8885	0.8874	0.8880	0.8876	0.8877	0.8880
Random crop 20%	0.7338	0.8224	0.8220	0.8110	0.7975	0.7898	0.7190	0.7584	0.7817	0.7936	**0.8464**
Random shift 10%	0.8909	0.9425	0.9487	0.9343	0.9302	0.9430	0.9304	0.9245	0.9294	0.9390	**0.9538**
Random shift 20%	0.8119	0.8295	0.8333	0.8780	0.8245	0.8536	0.8100	0.8124	0.8325	0.8377	**0.8941**

Table 5: The contribution of key features.

Models	Full **ABDH**		**ABDH** Without Feature 1		**ABDH** Without Feature 2		**ABDH** Without Feature 3	
Datasets/Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Set5	33.504676	0.964591	28.143444	0.811824	29.188589	0.838838	30.713594	0.868025
Set14	30.429662	0.948723	24.883460	0.671623	26.160978	0.704831	27.515278	0.749520

on the final generated target images, we use *Baboon* benchmark image from Set14 in Figure 4. The results shown in Table. 5 and Figure 4 show us how these three key features influence the quality and imperceptibility of the target image in qualitative and quantitative aspects.

Influence of Secret Embedding Amount

To further illustrate the effect of secret info embedding amount on the robustness and imperceptibility of the target image, we make the curve plots to show the quantitative results in Figure 5. We use the pixel-ratio to represent the amount of embedding secret info. It is defined as the ratio of pixels amount in secret image versus those in cover image. We can control the pixel-ratio by changing the size of the secret image. For example, if the size of the original cover image is 512×512, the size of the secret image is 256×256, then pixel-ratio is 0.25. In this experiment, we make the statistics in attacked dataset generated from COCO.

From the curve plots shown above, we can see *PSNR* and *SSIM* metrics decline with the increase of pixel-ratio. Under the noise attack or image compression, metrics are worse than the situations without attack, and also decline with the increase of pixel-ratio. It further proves the inherent contradiction between the embedded secret amount and robustness of the target image. So in the real applications, **ABDH** should make the trade-off between embedding amount, imperceptibility and attack robustness. This curve can tell the user the largest embedded secret capacity at certain imperceptibility and security level. So it is helpful for the user to choose the most suitable size of a secret image in real secure information transmission systems. For example, if the user wants to generate a target image with no less than 25dB *PSNR* and 0.97 *SSIM* versus cover image. Considering the noise attacks and image compression possibility, the largest embedded secret pixel-ratio should be less than 0.5.

Conclusion and Future Works

The good performance of **ABDH** derives from these factors.

- Discriminative network simulates detector, which helps to understand the sensitivity to semantic changes.
- Cycle discriminative model and inconsistent loss enhance the quality of generated target images.
- Mixed training dataset with noisy samples improves the robustness of **ABDH**. How to resist the tampering can be learned from attacked training samples.
- The introduction of the attention model further guides **ABDH** to find the inconspicuous areas of cover images, which is suitable for hiding secret info.

We have some initial results to prove that the introduction of attention mechanism helps the **ABDH** to find the relatively consistent and inconspicuous areas in videos. We will study the effectiveness of **ABDH** when hiding secret data to videos.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Baluja, S. 2017. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, 2069–2079.

Banitalebi, A.; Nader-Esfahani, S.; and Avanaki, A. N. 2018. Robust lsb watermarking optimized for local structural similarity. *arXiv preprint arXiv:1803.04617*.

Bansal, N.; Deolia, V. K.; Bansal, A.; and Pathak, P. 2016. Comparative analysis of digital watermarking techniques. In *Proceedings of the International Congress on Information and Communication Technology*, 105–115. Springer.

Bevilacqua, M.; Roumy, A.; Guillemot, C.; and Alberi-Morel, M. L. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding.

Das, U. K.; Samaddar, S. G.; and Keserwani, P. K. 2018. Digital forensic enabled image authentication using least significant bit (lsb) with tamper localization based hash function. In *Intelligent Communication and Computational Technologies*. Springer. 141–155.

Dong, S.; Zhang, R.; and Liu, J. 2018. Invisible steganography via generative adversarial network. *arXiv preprint arXiv:1807.08571*.

El Hossaini, A. E. A.; El Aroussi, M.; Jamali, K.; Mbarki, S.; and Wahbi, M. 2016. A new robust blind copyright protection scheme based on visual cryptography and steerable pyramid. *IJ Network Security* 18(2):250–262.

Ferdowsi, A., and Saad, W. 2017. Deep learning-based dynamic watermarking for secure signal authentication in the internet of things. *arXiv preprint arXiv:1711.01306*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Holub, V., and Fridrich, J. 2012. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, 234–239. IEEE.

Holub, V.; Fridrich, J.; and Denemark, T. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014(1):1.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. *arXiv preprint*.

Kinga, D., and Adam, J. B. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lu, J.; Wang, M.; Dai, J.; Huang, Q.; Li, L.; and Chang, C. 2015. Multiple watermark scheme based on dwt-dct quantization for medical images. *Journal of Information Hiding and Multimedia Signal Processing* 6(3):458–472.

Mun, S.-M.; Nam, S.-H.; Jang, H.-U.; Kim, D.; and Lee, H.-K. 2017. A robust blind watermarking using convolutional neural network. *arXiv preprint arXiv:1704.03248*.

Parah, S. A.; Sheikh, J. A.; Loan, N. A.; and Bhat, G. M. 2016. Robust and blind watermarking technique in dct domain using inter-block coefficient differencing. *Digital Signal Processing* 53:11–24.

Pevnỳ, T.; Filler, T.; and Bas, P. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*, 161–177. Springer.

Shi, H.; Dong, J.; Wang, W.; Qian, Y.; and Zhang, X. 2017. Ssgan: Secure steganography based on generative adversarial networks. In *Pacific Rim Conference on Multimedia*, 534–544. Springer.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Volkhonskiy, D.; Nazarov, I.; Borisenko, B.; and Burnaev, E. 2017. Steganographic generative adversarial networks. *arXiv preprint arXiv:1703.05502*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13(4):600–612.

Wolfgang, R. B., and Delp, E. J. 1996. A watermark for digital images. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, 219–222. IEEE.

Yi, Z.; Zhang, H. R.; Tan, P.; and Gong, M. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2868–2876.

Yu, C. 2016. Steganography of digital watermark based on artificial neural networks in image communication and intellectual property protection. *Neural Processing Letters* 44(2):307–316.

Zeyde, R.; Elad, M.; and Protter, M. 2010. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, 711–730. Springer.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding data with deep networks. *arXiv preprint arXiv:1807.09937*.