Multimedia Tools and Applications (2019) 78: 8559–8575
https://doi.org/10.1007/s11042-018-6951-z

Invisible steganography via generative adversarial networks

Ru Zhang[1] · **Shiqi Dong**[1] · **Jianyi Liu**[1]

Received: 24 July 2018 / Revised: 4 October 2018 / Accepted: 23 November 2018 /
Published online: 7 December 2018

Abstract
Nowadays, there are plenty of works introducing convolutional neural networks (CNNs) to the steganalysis and exceeding conventional steganalysis algorithms. These works have shown the improving potential of deep learning in information hiding domain. There are also several works based on deep learning to do image steganography, but these works still have problems in capacity, invisibility and security. In this paper, we propose a novel CNN architecture named as ISGAN to conceal a secret gray image into a color cover image on the sender side and exactly extract the secret image out on the receiver side. There are three contributions in our work: (i) we improve the invisibility by hiding the secret image only in the Y channel of the cover image; (ii) We introduce the generative adversarial networks to strengthen the security by minimizing the divergence between the empirical probability distributions of stego images and natural images. (iii) In order to associate with the human visual system better, we construct a mixed loss function which is more appropriate for steganography to generate more realistic stego images and reveal out more better secret images. Experiment results show that ISGAN can achieve start-of-art performances on LFW, PASCAL-VOC12 and ImageNet datasets.

Keywords Image steganography · Generative adversarial networks · Convolutional neural network

1 Introduction

Image steganography is the main content of information hiding. The sender conceal a secret message into a cover image, then get the container image called stego, and finish the secret message's transmission on the public channel by transferring the stego image. Then the receiver part of the transmission can reveal the secret message out. Steganalysis is an attack

✉ Shiqi Dong
shiqidong@bupt.edu.cn

Ru Zhang
zhangru@bupt.edu.cn

[1] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Xitucheng Road No. 10, HaiDian District, Beijing, China

to the steganography algorithm. The listener on the public channel intercept the image and analyze whether the image contains secret information. Since their proposed, steganography and steganalysis promote each other's progress.

Image steganography can be used into the transmission of secret information, watermark, copyright certification and many other applications. In general, we can measure a steganography algorithm by capacity, invisibility and security. The capacity is measured by bits-per-pixel (bpp) which means the average number of bits concealed into each pixel of the cover image. With the capacity becomes larger, the security and the invisibility become worse. The invisibility is measured by the similarity of the stego image and its corresponding cover image. The invisibility becomes better as the similarity going higher. The security is measured by whether the stego image can be recognized out from natural images by steganalysis algorithms. Correspondingly, there are two focused challenges constraining the steganography performance. The amount of hidden message alters the quality of stego images. The more message in it, the easier the stego image can be checked out. Another keypoint is the cover image itself. Concealing message into noisy, rich semantic region of the cover image yields less detectable perturbations than hiding into smooth region.

Nowadays, traditional steganography algorithms, such as S-UNIWARD [14], J-UNIWARD [14], conceal the secret information into cover images' spatial domain or transform domains by hand-crafted embedding algorithms successfully and get excellent invisibility and security. With the rise of deep learning in recent years, deep learning has become the hottest research method in computer vision and has been introduced into information hiding domain. Volkhonskiy et al. [25] proposed a steganography enhancement algorithm based on GAN, they concealed secret message into generated images with conventional algorithms and enhanced the security. But their generated images are warping in semantic, which will be drawn attention easily. Tang et al. [24] proposed an automatic steganographic distortion learning framework, their generator can find pixels which are suitable for embedding and conceal message into them, their discriminator is trained as a steganalyzer. With the adversarial training, the model can finish the steganography process. But this kind of method has low capacity and is less secure than conventional algorithms. Baluja [2] proposed a convolutional neural network based on the structure of encoder-decoder. The encoder network can conceal a secret image into a same size cover image successfully and the decoder network can reveal out the secret image completely. This method is different from other deep learning based models and conventional steganography algorithms, it has large capacity and strong invisibility. But stego images generated by this model is distorted in color and its security is bad. Inspired by Baluja's work, we proposed an invisible steganography via generative adversarial network named ISGAN. Our model can conceal a gray secret image into a color cover image with the same size, and our model has large capacity, strong invisibility and high security. Comparing with previous works, the main contributions of our work are as below:

1. In order to suppress the distortion of stego images, we select a new steganography position. We only embed and extract secret information in the Y channel of the cover image. The color information is all in Cr and Cb channels of the cover image and can be saved completely into stego images, so the invisibility is strengthened.
2. From the aspect of mathematics, the difference between the empirical probability distributions of stego images and natural images can be measured by the divergence. So we introduce the generative adversarial networks to increase the security throughout minimizing the divergence. In addition, we introduce several architectures from classic

computer vision tasks to fuse the cover image and the secret image together better and get faster training speed.

3. In order to fit the human visual system (HVS) better, we introduce the structure similarity index (SSIM) [27] and its variant to construct a mixed loss function. The mixed loss function helps to generate more realistic stego images and reveal out better secret images. This point is never considered by any previous deep-learning-based works in information hiding domain.

The rest of the paper is organized as follows. Section 2 discusses related works, Section 3 introduces architecture details of ISGAN and the mixed loss function. Section 4 gives details of different datasets, parameter settings, our experiment processes and results. Finally, Section 5 concludes the paper with relevant discussion.

2 Related works

Steganalysis There have been plenty of works using deep learning to do image steganalysis and got excellent performance. Qian et al. [17] proposed a CNN-based steganalysis model GNCNN, the model introduced the hand-crafted KV filter to extract residual noise and used the gaussian activation function to get more useful features. The performance of the GNCNN is inferior to the state-of-the-art hand-crafted feature set spatial rich model (SRM) [7] slightly. Based on GNCNN, Xu et al. [8] presented Batch Normalization [16] in to prevent the network falling into the local minima. XuNet was equipped with Tanh, 1×1 convolution, global average pooling, and got comparable performance to SRM [7]. Ye et al. [29] put forward YeNet which surpassed SRM and its several variants. YeNet used 30 hand-crafted filters from SRM to prepropose images, applied well-designed activation function named TLU and selection-channel module to strengthen features from rich texture region where is more suitable for hiding information. Zeng et al. [31] proposed a JPEG steganalysis model with less parameters than XuNet and got better performance than XuNet. These works have applied deep learning to steganalysis successfully, but there is still space for improvement.

Steganography Since its introduction, generative adversarial networks [10] have received more and more attention, achieved the state-of-art performance on tasks such as image generation, style transfer, speech synthesis and so on. The earliest application of deep learning to steganography was based on GAN. Volkhonskiy et al. [25] proposed a DCGAN-based [18] model SGAN. SGAN consists of a generator network for generating cover images, a discriminator network for discriminating generated images from real images and a steganalyzer network for steganalysis. Hiding information in cover images generated by SGAN is securer than in natural images. Shi et al. [22] proposed SSGAN based on WGAN [1], their work was similar to SGAN and got better outcome. However, stego images generated by models similar to SGAN and SSGAN are warping in semantic and are more easily to draw attention than natural images, although these models reduce the detection rate of steganalysis algorithms. Tang et al. [24] proposed an automatic steganographic distortion learning framework named as ASDL-GAN. The generator can translate a cover image into an embedding change probability matrix and the discriminator incorporates the XuNet architecture. In order to fit the optimal embedding simulator as well as propagate the gradient in back propagation, they proposed a ternary embedding simulator (TES) activation

function. ASDL-GAN can learn steganographic distortions automatically, but its performance is inferior to S-UNIWARD. Yang et al. [28] improved ASDL-GAN and achieved better performance than S-UNIWARD. They used Selection-Channel-Aware (SCA) [29] in generator as well as the U-Net framework [20] which is introduced from the medical images segmentation. However, ASDL-GAN still refers too many prior knowledge from conventional steganography algorithms and its capacity is small. Hayes [11] proposed a GAN-based model to hide a secret message into a cover image, and could reveal the secret message by his decoder successfully, but the invisibility is weak.

Baluja [2] designed a CNN model to conceal a color secret image into a color cover image yielding state-of-art performance. Atique et al. [19] proposed another encoder-decoder based model to finish the same steganography task (their secret images are gray images). This is a novel steganography method which gets rid of hand-crafted algorithms. It can learn how to merge the cover image and the secret image together automatically. But stego images generated by their models are distorted in color. As shown in Fig. 4, Atique's stego images are yellowing when compared with the corresponding cover images. And their stego images are easily recognized by well trained CNN-based steganalyzer [2] because of the large capacity. Inspired by works of Baluja and Atique, we improve each shortcoming and get ISGAN.

3 Our approach

The complete architecture of our model is shown in Fig. 1. In this section, the new steganography position is introduced firstly. Then we discuss about our design considerations on the basic model and show specfic details of the encoder and the decoder. Thirdly, we present why the generative adversarial networks can improve the security and details of the discriminator. Finally, we explain the motivation to construct the mixed loss function.

3.1 New steganography position

Works of Baluja [2] and Atique [19] have implemented the entire hiding and revealing procedure, while their stego images' color is distorted as shown in Fig. 4. To against this weakness, we select a new steganography position. As shown in Fig. 2, a color image in the RGB color space can be divided into R, G and B channels, and each channel contains both semantic information and color information. When converted to the YCrCb color space,

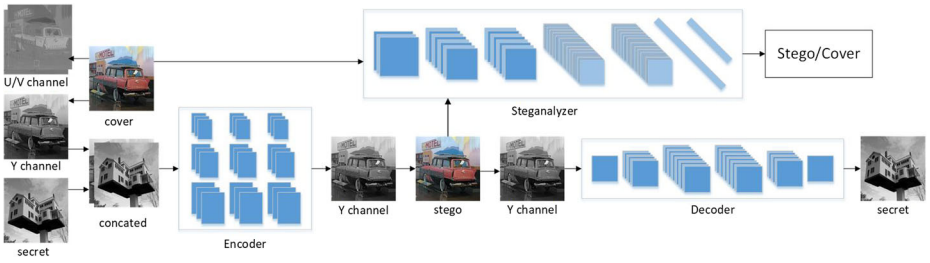


Fig. 1 The overall architecture. The encoder network conceals a gray secret image into the Y channel of a same size cover image, then the Y channel output by the encoder net and the U/V channels constitute the stego image. The decoder network reveals the secret image from the Y channel of the stego image. The steganalyzer network tries to distinguish stego images from cover images thus improving the overall architecture's security

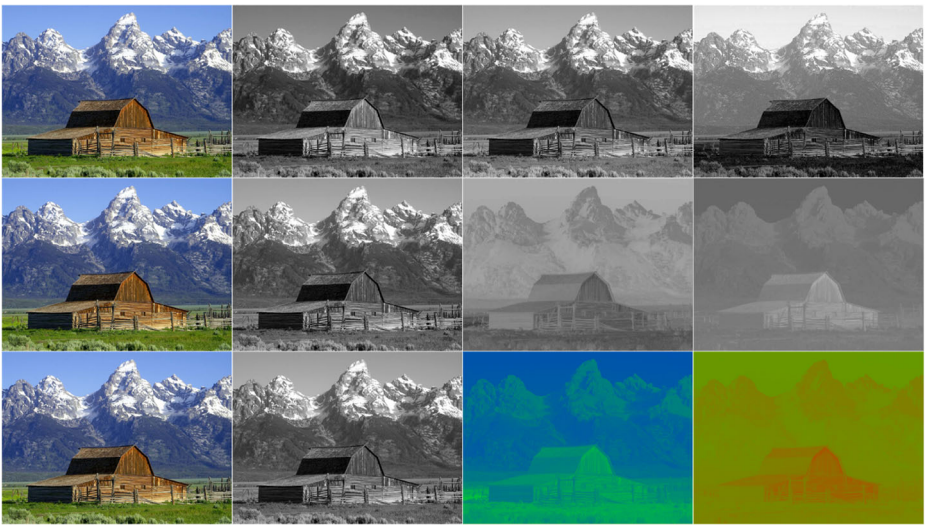


Fig. 2 Three images in the first column are original RGB color images. Three images in the right of the first row are R channel, G channel and B channel of the original image respectively saved as gray images, three channels all constitutes the luminance information and color information. Three images in the right of the second row are Y channel, Cr channel and Cb channel respectively saved as gray images, and three images in the right of the third row are also Y channel, Cr channel and Cb channel respectively from Wikipedia. We can see that the Y channel constitutes only the luminance information and semantic information, and the color information about chrominance and chroma are all in the Cr channel and the Cb channel

a color image can be divided into Y, Cr and Cb channels. The Y channel only contains part of semantic information, luminance information and no color information, Cr and Cb channels contain part of semantic information and all color information. To guarantee no color distortion, we conceal the secret image only in the Y channel and all color information are saved into the stego image. In addition, we select gray images as our secret images thus decreasing the secret information by $\frac{2}{3}$.

When embedding, the color image is converted to the YCrCb color space, then the Y channel and the gray secret image are concatenated together and then are input to the encoder network. After hiding, the encoder's output and the cover image's CrCb channels constitute the color stego image. When revealing, we get the revealed secret image through decoding the Y channel of the stego image. Besides, the transformation between the RGB color space and the YCrCb color space is just the weighted computation of three channels and doesn't affect the backpropagation. So we can finish this tranformation during the entire hiding and revealing process. The encoder-decoder architecture can be trained end-to-end, which is called as the basic model.

3.2 Basic model

Conventional or classic image stegnography are usually designed in a heuristic way. Generally, these algorithms decide whether to conceal information into a pixel of the cover image and how to conceal 1 bit information into a pixel. So the key of the classic steganography methods is well hand-crafted algorithms, but all of these algorithms need lots of expertise and this is very difficult for us. The best solution is to mix the secret image with the cover

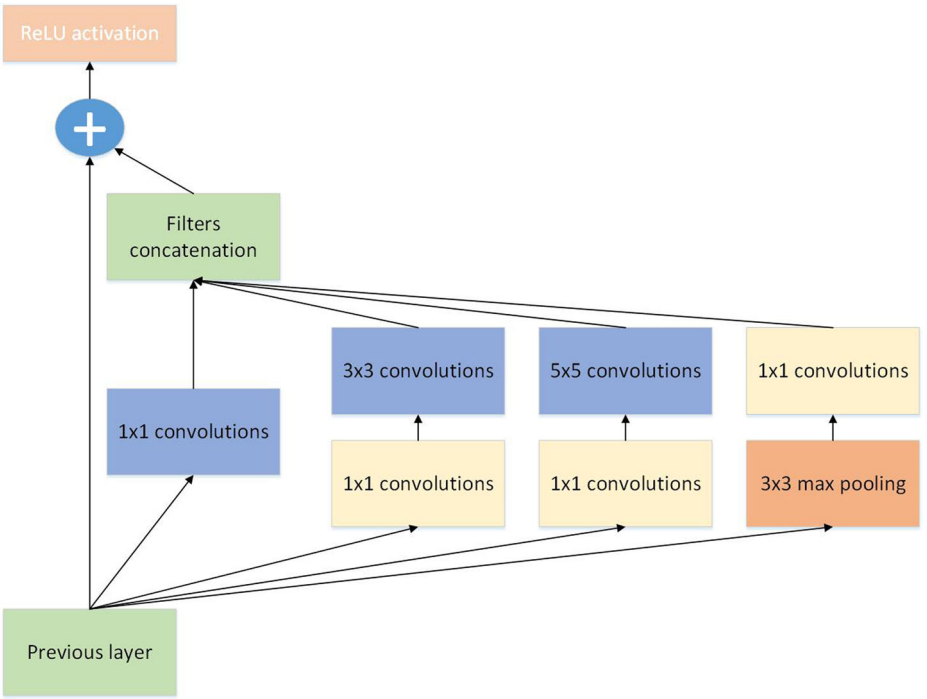


Fig. 3 The inception module with residual shortcut we use in our work

image very well without too much expertise. Deep learning, represented by convolutional neural networks, is a good way to achieve this exactly. What we need to do is to design the structure of the encoder and the decoder as described below.

Based on such a starting point, we introduce the inception module [23] in our encoder network. The inception module has excellent performance on the ImageNet classification task, which contains several convolution kernels with different kernel sizes as shown in Fig. 3. Such a model structure can fuse feature maps with different receptive field sizes

Table 1 Architecture details of the encoder network: ConvBlock1 represents 3 × 3 Conv+BN+LeakyReLU, ConvBlock2 represents 1 × 1 Conv+Tanh, InceptionBlock represents the inception module with residual shortcut as shown in Fig. 3

Layers	Process	Output size
Input	/	2 × 256 × 256
Layer 1	ConvBlock1	16 × 256 × 256
Layer 2	InceptionBlock	32 × 256 × 256
Layer 3	InceptionBlock	64 × 256 × 256
Layer 4	InceptionBlock	128 × 256 × 256
Layer 5	InceptionBlock	256 × 256 × 256
Layer 6	InceptionBlock	128 × 256 × 256
Layer 7	InceptionBlock	64 × 256 × 256
Layer 8	InceptionBlock	32 × 256 × 256
Layer 9	ConvBlock1	16 × 256 × 256
Output	ConvBlock2	1 × 256 × 256

Table 2 Architecture details of the decoder network: ConvBlock1 represents 3 × 3 Conv+BN+LeakyReLU, ConvBlock2 represents 1 × 1 Conv+Sigmoid

Layers	Process	Output size
Input	/	1 × 256 × 256
Layer 1	ConvBlock1	32 × 256 × 256
Layer 2	ConvBlock1	64 × 256 × 256
Layer 3	ConvBlock1	128 × 256 × 256
Layer 4	ConvBlock1	64 × 256 × 256
Layer 5	ConvBlock1	32 × 256 × 256
Output	ConvBlock2	1 × 256 × 256

very well. As shown in both residual networks [13] and batch normalization [16], a model with these modifications can achieve the performance with significantly fewer training steps comparing to its original version. So we introduce both residual module and batch normalization into the encoder network to speed up the training procedure. The detail structure of the encoder is described in Table 1. When using MSE as the metric on LFW dataset, we use our model to train for 30 epochs to get the performance Atique's model can achieve while training for 50 epochs.

On the other hand, we need a structure to reveal the secret image out automatically. So we use a fully convolutional network as the decoder network. Feature maps output by each convolutional layer have the same size. To speed up training, we add a batch normalization layer after each convolutional layer other than the last layer. Details of the decoder network are described in Table 2.

3.3 Our steganalyzer

Works of Baluja and Atique didn't consider the security problem, while the security is the keypoint in steganography. In our work, we want to take the steganalysis into account automatically throughout training the basic model.

Denoting $\mathcal{C}$ as the set of all cover images c, the selection of cover images from $\mathcal{C}$ can be described by a random variable c on $\mathcal{C}$ with probability distribution function (pdf) P. Assuming the cover images are selected with pdf P and embedded with a secret image which is chosen from its corresponding set, the set of all stego images is again a random variable s on $\mathcal{C}$ with pdf Q. The statistical detectability can be measured by the Kullback-Leibler divergence [3] shown in (1) or the Jensen-Shannon divergence shown in (2).

$$KL(P||Q) = \sum_{c \in \mathcal{C}} P(c) log \frac{P(c)}{Q(c)} \tag{1}$$

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

The KL divergence or the JS divergence is a very fundamental quantity because it provides bounds on the best possible steganalyzer one can build [4]. So the keypoint for us is how to decrease the divergence. The generative adversarial networks (GAN) are well-designed in theory to achieve this exactly. The objective of the original GAN is to minimize the JS divergence (2), a variant of the GAN is to minimize the KL divergence (1). The generator network G, which input is a noise z, tries to transform the input to a data sample which is similar to the real sample. The discriminator network D, which input is the real data or

the fake data generated by the generator network, determines the difference between the real and fake samples. D and G play a two-player minmax game with the value function (3).

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

Now we introduce the generative adversarial networks into our architecture. The basic model can finish the entire hiding and revealing process, so we use the basic model as the generator, and introduce a CNN-based steganalysis model as the discriminator and the steganalyzer. So the value function in our work becomes (4), where D represents the steganalyzer network, G represents the basic model, x, s and $G(x, s)$ represent the cover image, the secret image and the generated stego image respectively.

$$\min_G \max_D = E_{x\sim P(x)}[log D(x)] + E_{x\sim P(x),s\sim P(s)}[log(1 - D(G(x, s)))] \tag{4}$$

Xu et al. [8] studied the design of CNN structure specific for image steganalysis applications and proposed XuNet. XuNet embeds an absolute activation (ABS) in the first convolutional layer to improve the statistical modeling, applies the TanH activation function in early stages of networks to prevent overfitting, and adds batch normalization (BN) before each nonlinear activation layer. This well-designed CNN provides excellent detection performance in steganalysis. So we design our steganalyzer based on XuNet and adapt it to fit our stego images. In addition, we use the spatial pyramid pooling (SPP) module to replace the global average pooling layer. The spatial pyramid pooling (SPP) module [12] and its variants, which play a huge role in models for objection detection and semantic segmentation, break the limit of fully connected layers, so that images with arbitrary sizes can be input into convolutional networks with fully connected layers. On the other hand, the SPP module can extract more features from different receptive fields, thus improving the performance. Our steganalyzer's detail architecture is shown in Table 3.

3.4 Mixed loss function

In previous works, Baluja [2] used the mean square error (MSE) between the pixels of original images and the pixels of reconstructed images as the metric (1). Where c and s are the cover and secret images respectively, c' and s' are the stego and revealed secret images respectively, and β is how to weight their reconstruction errors. In particular, we should note that the error term $||c - c'||$ doesn't apply to the weights of the decoder network. On

Table 3 Architecture details of the steganalyzer network: ConvBlock1 represents 3 × 3 Conv+BN+LeakyReLU+AvgPool, ConvBlock2 represents 1 × 1 Conv+BN and ConvBlock3 represents 1 × 1 Conv+BN+LeakyReLU

SPPBlock contains a SPP module and the FC represents a fully connected layer

Layers	Process	Output size
Input	/	3 × 256 × 256
Layer 1	ConvBlock1	8 × 128 × 128
Layer 2	ConvBlock1	16 × 64 × 64
Layer 3	ConvBlock2	32 × 32 × 32
Layer 4	ConvBlock2	64 × 16 × 16
Layer 5	ConvBlock3	128 × 8 × 8
Layer 6	SPPBlock	2688 × 1
Layer 7	FC	128 × 1
Layer 8	FC	2 × 1

the other hand, both the encoder network and the decoder network receive the error signal $\beta||s - s'||$ for reconstructing the secret image.

$$L(c, c', s, s') = \|c - c'\| + \beta\|s - s'\| \tag{5}$$

However, the MSE just penalizes large error of two images' corresponding pixels but disregards the underlying structure in images. The human visual system (HVS) is more sensitive to luminance and color variations in texture-less regions. Zhao et al. [32] analyzed the importance of perceptually-motivated losses when the resulting image of image restoration tasks is evaluated by a human observer. They compared the performance of several losses and proposed a novel, differentiable error function. Inspired by their work, we introduce the structure similarity index (SSIM) [27] and its variant, the multi-scale structure similarity index (MS-SSIM) [26] into our metric.

The SSIM index separates the task of similarity measurement into three comparisons: luminance, contrast and structure. The luminance, contrast and structure similarity of two images are measured by (2), (3) and (4) respectively. Where μ_x and μ_y are pixel average of image x and image y, θ_x and θ_y are pixel deviation of image x and image y, and θ_{xy} is the standard variance of image x and y. In addition, C_1, C_2 and C_3 are constants included to avoid instability when denominators are close to zero. The total calculation method of SSIM is shown in (5), where $l > 0, m > 0, n > 0$ are parameters used to adjust the relative importance of three components. More detail introduction to SSIM can be found in [27]. The value range of the SSIM index is [0, 1]. The higher the index is, the more similar the two images are. So we use $1 - SSIM(x, y)$ in our loss function to measure the difference of two images. And the MS-SSIM [26] is an enhanced variant of the SSIM index, so we also introduce it into our loss function (We use $MSSIM$ in functions to represent MS-SSIM).

$$L(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{6}$$

$$C(x, y) = \frac{2\theta_x\theta_y + C_2}{\theta_x^2 + \theta_y^2 + C_2} \tag{7}$$

$$S(x, y) = \frac{\theta_{xy} + C_3}{\theta_x\theta_y + C_3} \tag{8}$$

$$SSIM(x, y) = [L(x, y)]^l \cdot [C(x, y)]^m \cdot [S(x, y)]^n \tag{9}$$

Considering pixel value differences and structure differences simultaneously, we put MSE, SSIM and MS-SSIM together. So, the metric for the basic steganography network in our framework is as below:

$$\begin{aligned} L(c, c') = \alpha(1 - SSIM(c, c')) \\ +(1 - \alpha)(1 - MSSIM(c, c')) \\ +\beta MSE(c, c') \end{aligned} \tag{10}$$

$$\begin{aligned} L(s, s') = \alpha(1 - SSIM(s, s')) \\ +(1 - \alpha)(1 - MSSIM(s, s')) \\ +\beta MSE(s, s') \end{aligned} \tag{11}$$

$$L(c, c', s, s') = L(c, c') + \gamma L(s, s') \tag{12}$$

Where α and β are hyperparameters to weigh influences of three metrics and γ is a hyperparameter to trade off the quality of stego images and revealed secret images. Experiment results in Section 4 will compare the performance of different loss functions.

4 Experiments and results

In this section, we'll introduce our experiment details and results. Firstly, the datasets we used are LFW [15], Pascal VOC 2012 [6] and ImageNet [5]. The Labeled Faces in the Wild (LFW) contains more than 13000 face images belonging to 1680 people collected from the web. 10k images were selected from LFW and constituted 5k cover-secret image pairs as our training set, others of LFW were as our validation set. Pascal VOC 2012 is a dataset designed for object detection and semantic segmentation, we selected 16k images randomly to constitute 8k cover-secret image pairs as our training set and selected 5k images from the remaining part as our validation set. To further verify our model's performance on the big dataset, we did similar experiments on a subset of the ImageNet. Limited by the computing power, we only used the validation set of ImageNet as our training set which contains 50k images, these images constituted 25k cover-secret image pairs randomly. Then we selected 30k images from the test set of ImageNet as our validation set.

We used SSIM [27], Peak Signal to Noise Ration (PSNR) as metrics to measure our model's performance. It is widely accepted that the PSNR doesn't correlate well with the human's perception of image quality [30], so we just used it as a reference. In addition, we designed a CNN-based steganalyzer specially to measure our model's security.

All settings of our model on three datasets were the same. All parameters of our model were initialized by the Xavier initialization [9] and the initial learning rate was set as 1e-4 and was descended during training after 20 epochs. The batch size was set as 4 limited by the computing power, and we used Adam to optimize our basic model. After several attempts, we set α, β and γ of the loss function as 0.5, 0.3 and 0.85 respectively, which can trade off the quality of stego images and revealed secret images very well. Because our secret message is an image, so we don't need to reveal out the secret image completely. Certainly, you can set γ higher if you want better revealed secret images. The size of all images we used is 256×256, and the capacity of our model is 8bpp (it is equivalent to that we hide a pixel (8 bits) in a pixel).

As shown in Table 4, we do several experiments with different loss functions on the LFW, the result demonstrates that our proposed mixed loss function is superior to others. Table 5 describes final results of our model on three datasets, we can see that the invisibility of our model get a little improvement, while our model's performance is superior to Atique's work intuitively as shown in Figs. 4, 5 and 6. Stego images generated by our model are complete similar to corresponding cover images in semantic and color, this is not reflected by SSIM. On the training set, the average SSIM index between stego images generated by

Table 4 We use several loss functions to train our basic model on LFW for 50 epochs

Loss function	Stego-cover PSNR (db)	Revealed-secret PSNR (db)	Stego-cover SSIM	Revealed-secret SSIM
MSE	27.97	26.30	0.8592	0.8391
SSIM	21.71	22.76	0.8877	0.8466
MSE+SSIM	27.12	26.71	0.8921	0.8805
MSE+MS-SSIM	23.92	25.97	0.8287	0.8832
MSE+SSIM+MS-SSIM	26.72	25.97	0.9305	0.9160

$MSE + SSIM$ represents a mixed loss of MSE and SSIM, others are similar, and $revealed$ represents revealed secret images. The results show the mixed loss of MSE, SSIM and MS-SSIM is superior than others

Table 5 We can see that the SSIM index between stego images and their corresponding cover images of ISGAN is higher than our basic model and Atique's work [19]

Model	Cover image	Secret image	Stego-cover PSNR (db)	Revealed-secret PSNR (db)	Stego-cover SSIM	Revealed-secret SSIM
Atique's model	LFW	LFW	33.7	39.9	0.95	
Basic model	LFW	LFW	34.28	33.53	0.9529	0.9453
ISGAN	LFW	LFW	34.63	33.63	0.9573	0.9429
Atique's model	ImageNet	ImageNet	32.9	36.6	0.96	0.96
Basic model	ImageNet	ImageNet	34.57	33.53	0.9634	0.9510
ISGAN	ImageNet	ImageNet	34.89	33.42	0.9681	0.9474
Atique's model	PASCAL-VOC12	PASCAL-VOC12	33.7	35.9	0.96	0.95
Basic model	PASCAL-VOC12	PASCAL-VOC12	33.79	33.47	0.9617	0.9475
ISGAN	PASCAL-VOC12	PASCAL-VOC12	34.49	33.31	0.9661	0.9467
Atique's model	PASCAL-VOC12	LFW	33.8	37.7	0.96	0.95
Basic model	PASCAL-VOC12	LFW	33.85	37.68	0.9612	0.9503
ISGAN	PASCAL-VOC12	LFW	34.45	37.59	0.9647	0.9495
ISGAN	ImageNet	PASCAL-VOC12	34.57	36.58	0.9652	0.9495

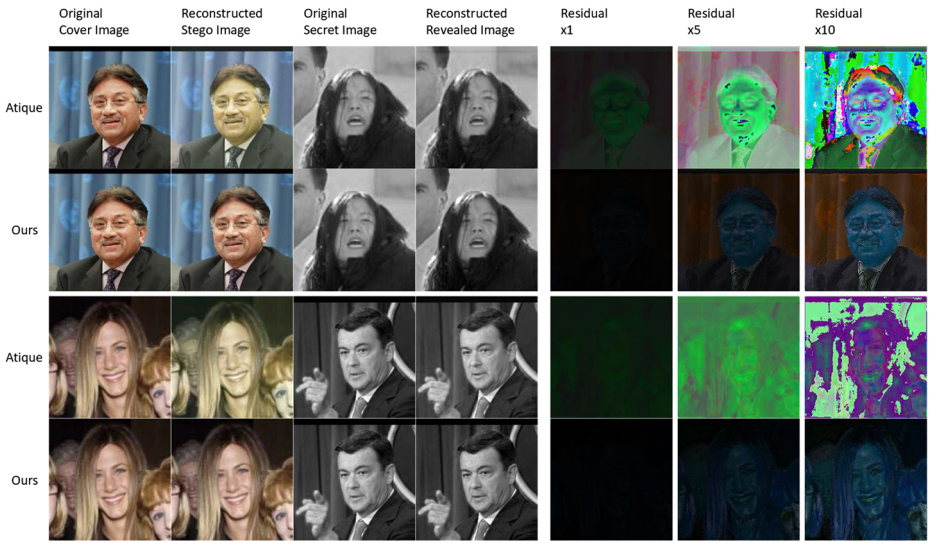


Fig. 4 Two examples on LFW. The results show that our stego images are almost same as cover images, while Atique's stego images are yellowing. By analyzing residuals between stego images and cover images, we can see that our stego images are more similar to cover images than Atique's results

our model and their corresponding cover images is more than 0.985, and the average SSIM index between revealed images and their corresponding secret images is more than 0.97. In practice, we can use several cover images to conceal one secret image and choose the best stego image to transfer on the Internet.

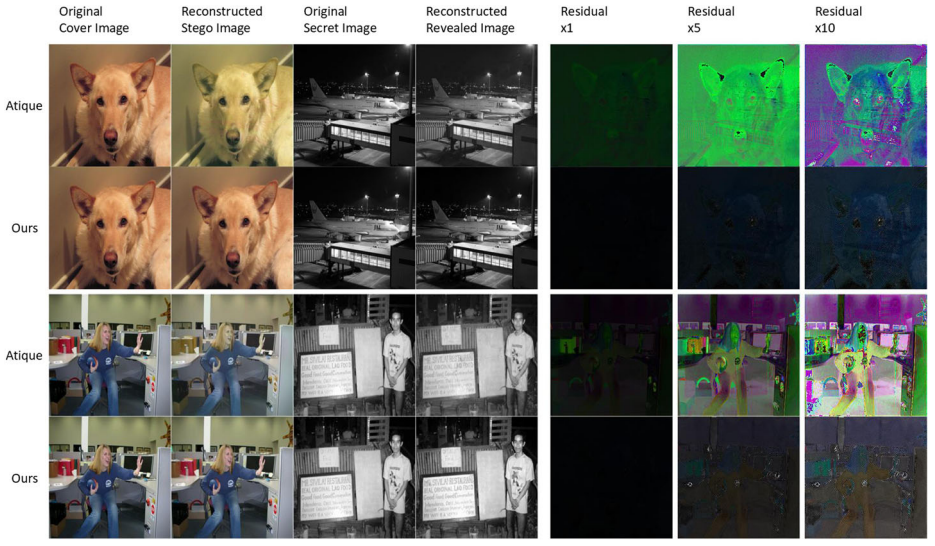


Fig. 5 Two examples on Pascal VOC12. We can see that our stego images are almost same as cover images, while Atique's stego images are yellowing. By analyzing residuals between stego images and cover images, we can even distinguish the outline of secret images from Atique's residual images, while our residual images are blurrier

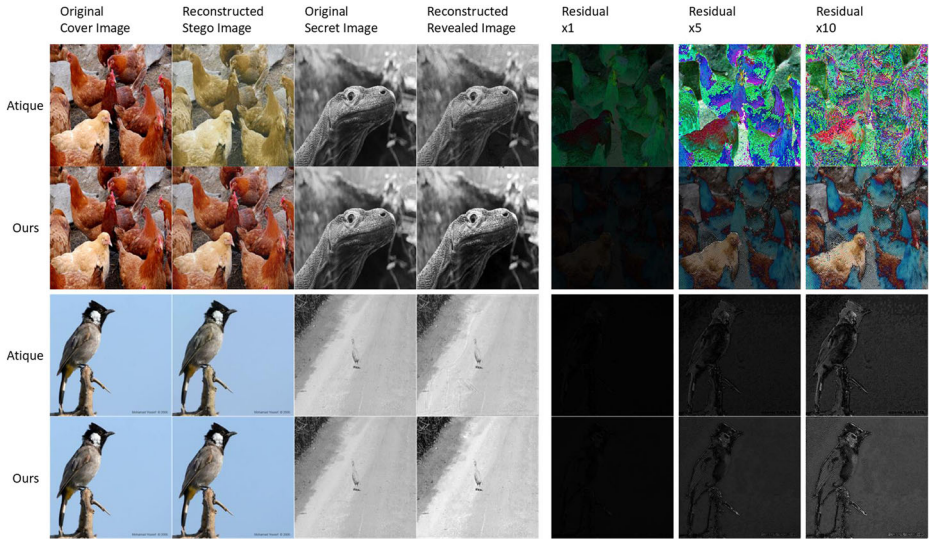


Fig. 6 Two examples on ImageNet. The results show that our stego images are almost same as cover images, while Atique's stego images are yellowing. Residual images between stego images and cover images show that our stego images are more similar to cover images than Atique's results

On the other hand, by analyzing the detail difference between cover images and stego images, we can see that our residual images are darker than Atique's, which means that our stego images are more similar to cover images and ISGAN has stronger invisibility. Additionally, from Atique's residual images we can even distinguish secret images' outline, while our residual images are blurrier. So these residual images can also prove that our ISGAN is securer.

When training ISGAN, we referred some tricks from previous works [21]. We flipped labels when training our basic model, replaced the ReLU activation function by the LeakyReLU function, optimized the generator by Adam, optimized the steganalyzer by SGD and applied the L2 normalization to inhibit overfitting. These tricks helped us to speed up training and get better results.

To prove the improvement of the security produced by generative adversarial networks, we designed a new experiment. We used a well-trained basic model to generate 5000 stego images on LFW. These 5000 stego images and their corresponding cover images constituted a tiny dataset. We designed a new CNN-based model as a binary classifier to train on the tiny dataset. After training, we used this model to recognize stego images out from another tiny dataset which contains 2000 stego images generated by ISGAN and their corresponding cover images. Similar experiments were done on the other two datasets. The results can be seen from Table 6. ISGAN strengthens indeed the security of our basic model. And with the training going, the security of ISGAN is improving slowly.

5 Discussion and conclusion

Figure 7 shows the difference between revealed images and their corresponding secret images. It shows that this kind of model cannot reveal out secret images completely. This is

Table 6 Accuracy of CNN-based steganalysis model on tiny-datasets generated by basic model and ISGAN training for different epochs

Dataset	Basic Model	ISGAN (50)	ISGAN (100)	ISGAN (150)
LFW	0.8305	0.8059	0.7887	0.7825
Pascal-VOC12	0.7953	0.769	0.756	0.7438
ImageNet	0.7814	0.7655	0.7462	0.7360

Along with the training going, we can see that the security of ISGAN is improving slowly

accepted as the information in the secret image is very redundant. However, it is unsuitable for tasks which need to reveal the secret information out completely.

As we described before, ISGAN can conceal a gray secret image into a color cover image with the same size excellently and generate stego images which are almost the same as cover images in semantic and color. By means of the adversarial training, the security is improved. In addition, experiment results demonstrate that our mixed loss function based on SSIM can achieve the state-of-art performance on the steganography task.

In addition, our steganography is done in the spatial domain and stego images must be lossless, otherwise some parts of the secret image will be lost. There may be methods to address this problem. It doesn't matter if the stego image is sightly lossy since the secret image is inherently redundant. Some noise can be added into the stego images to simulate the image loss caused by the transmission during training. Then our decoder network should be modified to fit both the revealing process and the image enhancement process together. In our future work, we'll try to deal with this problem and improve our model's robustness.

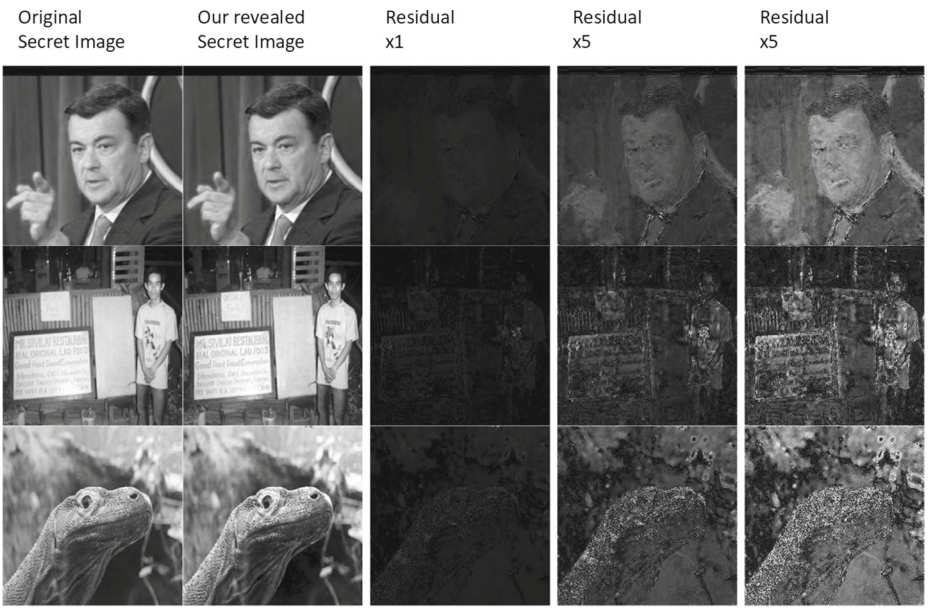


Fig. 7 Secret images' residual image on three datasets. There are differences between original secret images and our revealed secret images, which means that ISGAN is a lossy steganography

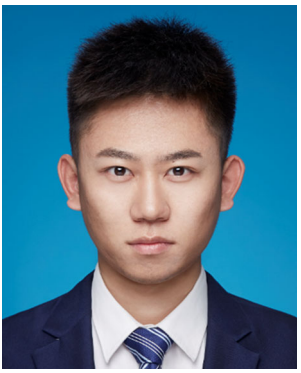
Acknowledgements This work was supported by the National key Research and Development Program of China(No.2016YFB0800404) and the NSF of China(U1636112,U1636212).

References

1. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN. In: Proceedings of the 34th international conference on machine learning (ICML), pp 214–223
2. Baluja S (2017) Hiding images in plain sight: Deep steganography. In: Proceedings of advances in neural information processing systems 30 (NIPS), pp 2069–2079
3. Cachin C (1998) An information-theoretic model for steganography. In: Aucsmith D (ed) Information hiding, 2nd international workshop, volume 1525 of lecture notes in computer science, pp 306–318
4. Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York
5. Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 248–255
6. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88:303–338
7. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Trans Inf Forensics Secur 7(3):868–882
8. Guanshuo XU (2017) Deep convolutional neural network to detect JUNIWARD. In: Proceedings of 5th ACM workshop Inf Hiding Multimedia Secur (IH&MMSec), p 6773
9. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res 9:249–256
10. Goodfellow I, Pouget-Abadie J, Mirza M, Bing XU, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Proces 27(NIPS):2672–2680
11. Hayes J, Danezis G (2017) Generating steganographic images via adversarial training. In: Proceedings of advances in neural information processing systems 30 (NIPS), pp 1954–1963
12. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
13. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
14. Holub V, Fridrich J, Denemark T (2014) Universal distortion function for steganography in an arbitrary domain. EURASIP J Inf Secur 2014(1):1–13
15. Huang GB, Mattar M, Lee H, Learned-Miller E (2007) Labeled faces in the wild: a database for studying face recognition in unconstrained environments, University of Massachusetts, Amherst. Technical Report 07-49, October
16. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning (ICML), vol 37, pp 448–456
17. Qian Y, Dong J, Wang W, Tan T (2015) Deep learning for steganalysis via convolutional neural networks. Proc SPIE Int Soc Opt Eng 9409:94090J–94090J-10
18. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of international conference on learning representations (ICLR)
19. Rehman AU, Rahim R, Nadeem S, Hussain SU (2017) End-to-end trained cnn encoder-decoder networks for image steganography. arXiv:1711.07201
20. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Proceedings of international conference on medical image computing and computer-assisted intervention, pp 234–241

21. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Xi C (2016) Improved techniques for training GANs. In: Proceedings of advances in neural information processing systems 29 (NIPS), pp 2234–2242
22. Shi H, Dong J, Wang W, Qian Y, Zhang X (2018) SSGAN: secure steganography based on generative adversarial networks. In: Zeng B, Huang Q, El Saddik A, Li H, Jiang S, Fan X (eds) Advances in multimedia information processing – PCM 2017. PCM 2017 lecture notes in computer science, vol 10735. Springer, Cham
23. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 2818–2826
24. Tang W, Tan S, Li B, Huang J (2017) Automatic steganographic distortion learning using a generative adversarial network. IEEE Signal Process Lett 24(10):1547–1551
25. Volkhonskiy D, Borisenko B, Burnaev E (2016) Generative adversarial networks for image steganography. In ICLR 2016 Open Review
26. Wang Z, Simoncelli EP, Bovik AC (2003) Multiscale structural similarity for image quality assessment. In: The 37th asilomar conference on signals, system and computers, vol 2, pp 1398–1402
27. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13:600–612
28. Yang J et al (2018) Spatial image steganography based on generative adversarial network. arXiv:1804.07939
29. Ye J, Ni J, Yi Y (2017) Deep learning hierarchical representations for image steganalysis. IEEE Trans Inf Forensics Secur 12:2545–2557
30. Zhang L, Zhang L, Mou X, Zhang D (2012) A comprehensive evaluation of full reference image quality assessment algorithms. In: Proceedings of the 19th IEEE international conference on image processing, pp 1477–1480
31. Zeng J et al (2018) Large-scale JPEG steganalysis using hybrid deep-learning framework. IEEE Trans Inf Forensics Secur 13:1200–1214
32. Zhao H, Gallo O, Frosio I et al (2017) Loss functions for image restoration with neural networks. IEEE Trans Comput Imaging 3:47–57

Ru Zhang received the Ph.D. degree from the School Of Computer Science, Beijing Institute of Technology, China, in 2003. She is currently an Associate Professor with the Beijing University of Posts and Telecommunications. Her research interests cover multimedia security, steganography and watermarking, and vulnerability analysis of industrial control system.



Shiqi Dong received the B.S. degree from Hebei University in 2015. He is currently working toward the M.S. degree at the Beijing University of Posts and Telecommunications. His research interests include steganography, pose estimation and computer vision.

Jianyi Liu received the B.S. degree from Xi'an University of Posts and Telecommunications in 2000 and the Ph.D. degree in engineering from the Beijing University of Posts and Telecommunications in 2005. His current research interests cover information security, natural language processing, and big data analysis.