

# Two-Step Image-in-Image Steganography via GAN

Guanzhong Wu, South China University of Technology, China

Xiangyu Yu, South China University of Technology, China\*

Hui Liang, South China University of Technology, China

Minting Li, South China University of Technology, China

## ABSTRACT

Recently, the convolutional neural network has been introduced to information hiding, and deep network has shown great potential in steganography. However, one drawback of deep network is that it's sensitive to small fluctuations. In previous works, the encoder-decoder structure was trained end-to-end, but in practice, encoder and decoder should be used separately. Therefore, end-to-end-trained steganography networks are vulnerable to fluctuations, and the secret decoded from those networks suffers from unpleasant noise. In this work, the authors present an image-in-image steganography method called TISGAN to achieve better results, both in image quality and security. In particular, this research divides the training process into two parts. Moreover, perceptual loss is applied to encoder to improve security in the work. This article also appends a denoising structure to the end of decoder to achieve better image quality. Finally, the adversarial structure with useful techniques employed is also used in secret revealed process.

## KEYWORDS

Convolution Neural Network, Deep Learning, Generative Adversarial Networks, Steganography

## 1. INTRODUCTION

Steganography is a technology in the field of information hiding. It is defined as concealing a file, message, image, or video within another carrier medium. In this paper, we focus on steganography in image, one of the most commonly used medium. In an image steganography case (Abadi & Andersen, 2016), Alice encodes a secret message in an image and send it through the public channel to Bob, who can decode the image and extract secret message from it. But Eve can't distinguish whether the image Alice sent contains secret or not. Eve's detecting stego image is called steganalysis, a technology against steganography.

Generally, security, robustness, and capacity are three metrics to measure a steganography algorithm. Security means the difficulty for a steganalysis algorithm to detect an stego image with certain message. Robustness is the ability of the hidden message to remain undamaged even if the stego media undergoes transformation, sharpening, linear and non-linear filtering, scaling, blurring, cropping and various other techniques. Capacity is the average amount of information hidden in a unit, which is usually measured by bit-per-pixel.

DOI: 10.4018/IJDCF.295814

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

There are a wide variety of steganography algorithms for different applications. At the very beginning, least significant bit(LSB) method changes the least significant bit of a pixel to hide secret message. Variants of LSB methods are described in (Wolfgang & Delp, 1996). Although cover image using LSB methods can't be found visually, it can be easily detected by analyzing the statistics of image. To solve this problem, steganography algorithms such as HUGO (Pevný et al., 2010), HILL (Li et al., 2014), WOW (Holub & Fridrich, 2012) and S-UNIWARD (Holub et al., 2014) hide secret information by minimizing a distortion metric.

With rapid development of deep convolution neural networks, generative adversarial network(GAN) (Goodfellow et al., 2014) was proposed and is widely used in computer vision. There are lots of works that focus on improving GAN, especially on stabilizing the training process. WGAN (Arjovsky et al., 2017) introduced Wasserstein distance that can accurately measure the distance between two different distributions instead of Kullback–Leibler(KL) divergence. Later on, Gulrajani et al. (2017) used gradient penalty instead of weight clipping in WGAN. Spectral normalization (Miyato et al., 2018) further improves the performance. Jolicœur-Martineau (2018) used relativistic discriminator not only to judge whether a sample is real or not, but also to increase the probability that generated data are real. Except improvement in discriminator, Heusel et al. (2017) proposed different learning rate for discriminator and generator is also useful. In our work, many techniques are adopted to stabilize the training process. GAN is also introduced to steganography these years. As a pioneer, Volkhonskiy et al. (2017) proposed SGAN. They used GAN to generate some images as cover, and hid secret in the generated images. Based on their work, Shi et al. (2017) used WGAN to improve the quality of cover images, decreasing the artifacts. Conservatively, ASDL-GAN (Yang et al., 2018), which used neural network to generate a distortion metric, was proposed. Recently, Rehman et al. (2017) proposed the end-to-end training CNN, an encode-decode structure. This kind of algorithm is a blind steganography method in which receiver is unnecessary to have access to the original image, so it is automatic and very convenient to use in practice. With such advantages, the encode-decode structure has been widely used ever since. A deep steganography method that can hide an image to another image is proposed by Baluja (2017). Based on his work, Dong et al. (2018) introduced a network that hid a grey-scale image to a colorful image with GAN. However, the secret images revealed by their work have low quality, suffering from severe noise.

As for image steganography, perceptual-driven approaches are also suitable to improve the visual quality. Based on the idea of maximizing perceptual similarity, Johnson et al. (2016) proposed perceptual loss to enhance the visual quality by minimizing the error in a feature space. To remove the disturbing noise, BM3D (Dabov et al., 2007) is a widely used blind denoising algorithm. Jain and Seung (2009) proposed to use convolutional neural network for image denoising and claimed that deep methods are even better than traditional methods. Zhang et al. (2016) proposed a Deep CNN for image denoising and achieved state-of-the-art performance.

Moreover, end-to-end training is adopted in these network, where secret image is decoded from the stego image without quantization, but it's improper in practice. To solve this problem, we introduced two-step image-in-image steganography via GAN, a new trainable framework for image-in-image steganography, to solve problems existing in their work. Following Dong et al. (2018), we hide a gray-scale image in a colorful image. Compared with previous works, our method has several modifications as below:

1. In order to reduce the impact of the quantification, which the end-to-end network can't handle well, the training process is divided into two parts, one for encoding and the other for decoding.
2. To get a clear secret image revealed from the decoder, we must reduce the impact of noise that severely affects the decoded image. The structure of original decoder is too simple to remove the noise. Therefore, we append a denoising network to the end of decoder and call it denoiser.
3. According to GAN theory, the noise distribution of the model can be measured. Besides using the steganalyzer in the encoder part, we also add a discriminator to judge whether an image is

the original secret image or the revealed one. In this way, noise distribution can be measured roughly and the effect of noise can be further relieved.

4. We introduce perceptual loss to the encoder loss function. Perceptual loss can significantly improve the visual quality of stego image, and also enhance security and robustness. As we know, perceptual loss has never been used in steganography.

The rest of the paper are organized as followed. Section 2 describes the theory and implementation of our method. Section 3 analyses the experimental results. A conclusion is arrived in Section 4.

## 2. THE PROPOSED METHOD

The main purpose of this paper is to improve the quality of stego images and revealed images. In this section, our proposed framework and a novel training method are introduced first. Then we discuss an improvement on the loss function. Finally, some techniques including spectral normalization and relativistic loss we use in training GAN are mentioned.

### 2.1 Network Structure

The network structure is shown in Figure 1. Inspired by ISGAN (Dong et al., 2018), our basic model has similar structure. The detail of encoder and decoder is shown in Figure 2. And we use XuNet (Xu et al., 2016) as the steganalizer.

Based on the basic model, we make three modifications to the whole network to further improve the quality of revealed secret image:

- I. Append a Denoiser after the decoder to remove the noise.
- II. Divided the end-to-end network into two parts to solve quantization problem.
- III. Add a Discriminator to improve the image quality in some details.

Figure 1. The whole framework

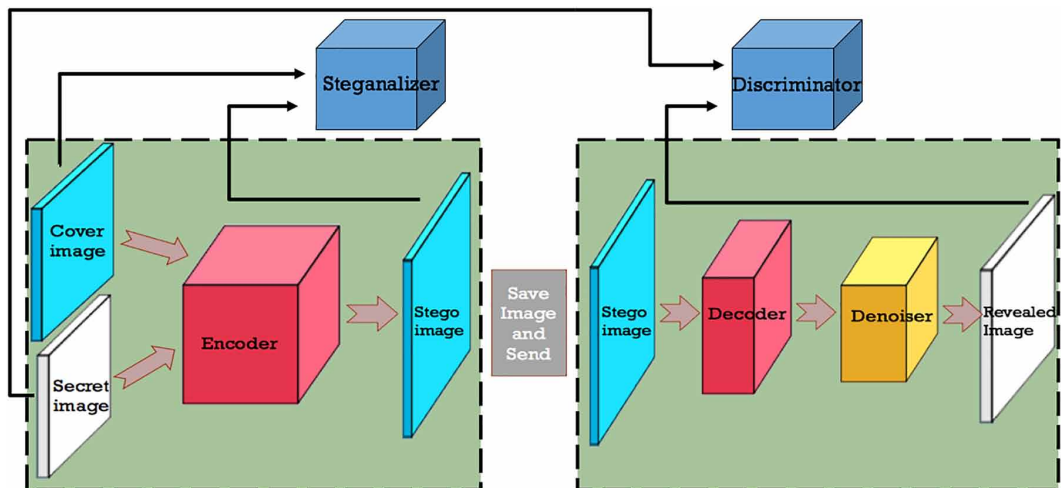


Figure 2. The details of basic model. (a) Encoder structure. (b) Decoder structure. (c) The Inception Block in encoder.

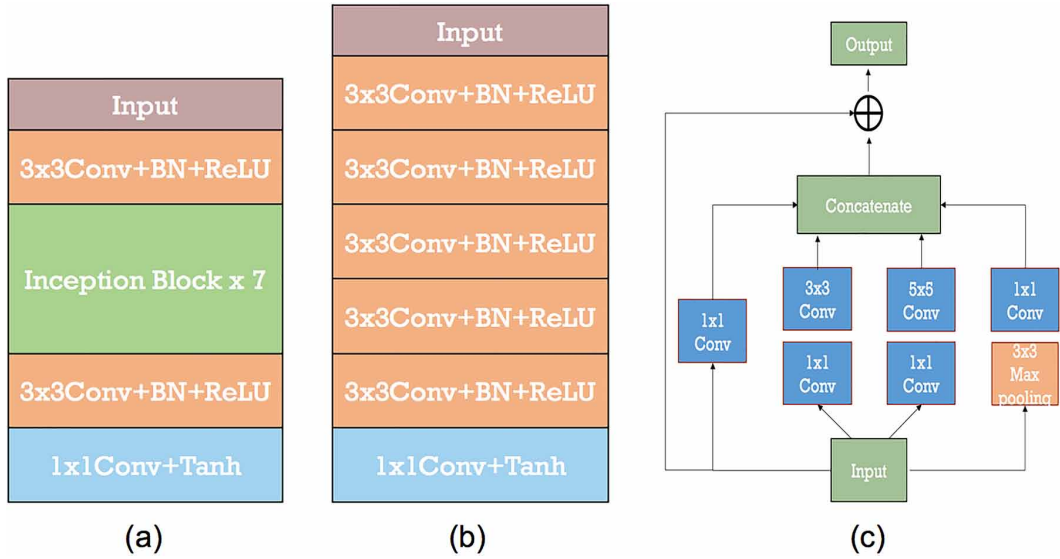
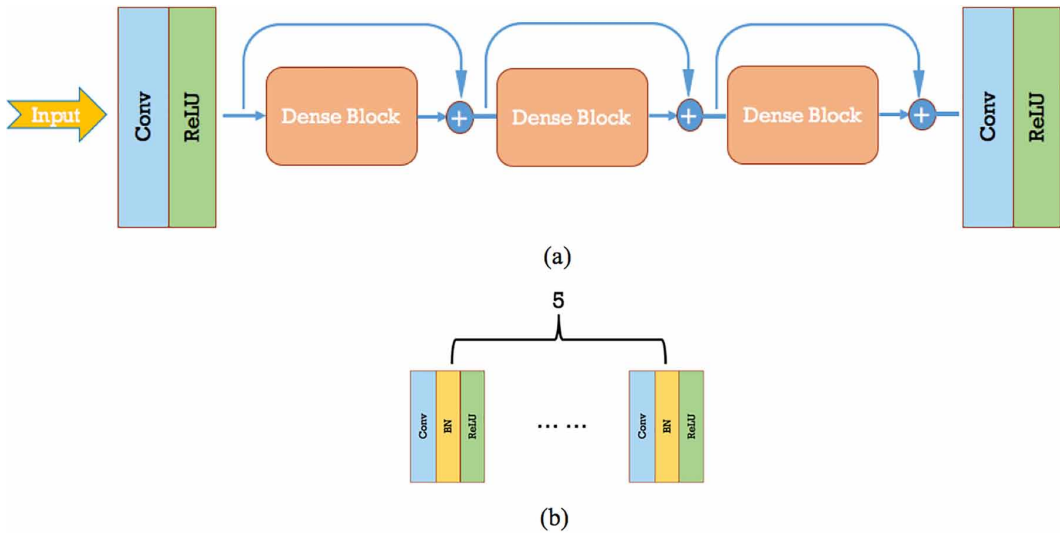


Figure 3. The details of denoiser. (a)Denoiser structure. Compared to DnCNN, we add the residual connection and decrease the depth. (b) Dense block structure.



As depicted in Figure 5 in Section 3.3, revealed images suffer from serious noise. Thus, it's necessary to add a structure to protect the revealed image from the noise. DnCNN (Zhang et al., 2016) has excellent performance on blind denoise task, which contains several convolution kernels with different kernel sizes. We modify the DnCNN as shown in Figure 3(a). And the dense block in DnCNN is shown in Figure 3(b).

To get a better revealed image, we also modify the reveal network as a GAN structure. Since the ResNet (He et al., 2015) structure performs well in many classification task, ResNet64 network is used as our discriminator to distinguish the revealed image and original secret image. Other technique benefits to training GAN also appears in our work, which will be described in Section 2.4.

## 2.2 Training Methods

In this section, a two-step training method is introduced. The first part is the hiding part, and the second part is revealing part. In previous work, encoder and decoder are always trained end-to-end as a whole, which means the decoder takes the stego image without quantization as its input. But it is improper. In steganography, the sender must keep the encoder and hide secret in the cover, and send the stegos through the public channel. The receiver keeps the decoder and reveals the secret from the stegos. The encoder and decoder must be used apart, which is impossible to be an end-to-end network. When the stego is saved as a picture, the datatype is “uint8”, but in the end-to-end network, the stego from the encoder should be considered as a 3-channel feature map with the datatype “float32”. Therefore, the saving process changes the datatype and a little message is lost. That’s the reason why the whole network can’t be trained end-to-end. Experiments in Section 3 will prove our opinion.

The hiding part, the purpose of whom is to train the encoder, is similar to the training process in ISGAN (Dong et al., 2018). If the encoder is trained alone, the stegos may be the same as the cover images without any secret information. Thus, encoder and decoder must be trained together in the hiding part to ensure that secret is in the stegos and can be revealed. To ensure that the stego image is secure from steganalysis, a steganalyzer, XuNet (Xu et al., 2016), is adopted in our network. During training, encoder would try to produce stego images that can fool the steganalyzer and further produce more secure stego images.

The revealing part aims at gaining the secret from stegos. As mentioned above, the decoder trained in the hiding part can not reveal the secret well because it was trained for the end-to-end network but not for the “uint8” picture. Therefore, the decoder should be finetuned with quantized stegos. We train the denoiser and finetune the decoder with the assistant of discriminator at the same time.

## 2.3 Perceptual Loss

In previous works, Dong et al. (2018) used mean square error(MSE), structure similarity index(SSIM) (Wang et al., 2004a) and its variant, the multi-scale structure similarity(MS-SSIM) (Wang, et al., 2004b) between cover images and stego images as the metrics, and the loss function of encoder is shown as follows:

$$L_{content}(c, c') = \alpha(1 - SSIM(c, c')) + (1 - \alpha)(1 - MSSIM(c, c')) + \beta MSE(c, c') \quad (1)$$

where  $c$  and  $c'$  are the cover and stego image, and  $\alpha$  and  $\beta$  are coefficients. We set  $\alpha = 0.5$  and  $\beta = 0.7$  in experiment.

Based on MSE and SSIM, a more effective loss, perceptual loss (Johnson et al., 2016) is added in our work. Perceptual loss is defined on the convolutional layers of a deep network, where the distance between two features is minimized. Wang et al. (2018) proved that using the features before the activation layers will be more efficient than after activation. As a steganalyzer, XuNet (Xu et al., 2016) accepts the original cover image and stego image as input, and outputs a scalar as a symbol whether it is an image contains secret message. With perceptual loss, stego image not only gets a close score to original image but also has similar feature maps on the convolution layer of steganalyzer. It means that it is difficult for steganalyzer to tell the difference between stego image and the security will be enhanced. Perceptual loss is measured as follows:

$$L_{percep}^{\phi,j} = \frac{1}{C_j W_j H_j} \left\| \phi_j(I_{stego}) - \phi_j(I_{cover}) \right\|_2^2 \quad (2)$$

where  $\phi_j$  denotes the feature maps obtained in  $j$ -th convolution in steganalyzer. And  $C_j$ ,  $W_j$ ,  $H_j$  denote the channels, width and height in the  $j$ -th convolution. Therefore, the total loss function of encoder is as follows:

$$L(c, c') = L_{content} + \lambda L_{percep} + \eta L_{adversarial} \quad (3)$$

where  $L_{adversarial}$  is the adversarial loss computed by the steganalyzer with Equation (4).  $\lambda$  and  $\eta$  are coefficients to balance the different loss terms.

## 2.4 Techniques to Stabilize GAN Training

Some techniques are adopted in the GAN training process. Different from the standard discriminator, spectral normalization is applied in our method based on SNGAN (Miyato et al., 2018). This operation constrains the Lipschitz constant of the discriminator that can measure the distance between two different data distributions accurately, and stabilize the training of GANs.

Based on Relativistic GAN (Jolicoeur-Martineau, 2018), standard loss of discriminator is adapted to relativistic loss. It helps the discriminator estimate how much more realistic a real sample is compared to a fake one. In this way, the loss function is described as follows:

$$L_G = -E_{xr}[\log(1 - D(x_r, x_f))] - E_{xf}[\log D(x_r, x_f)] \quad (4)$$

$$L_D = -E_{xr}[\log D(x_r, x_f)] - E_{xf}[1 - \log D(x_r, x_f)] \quad (5)$$

In Equation (4) and (5),  $E_{xr}[\bullet]$  represents the expectation of real images while  $E_{xf}[\bullet]$  represents that of fake images in a mini-batch. It's noticed that  $D(x_r, x_f) = \sigma(C(x_r) - E_{xf}[C(x_f)])$ , where  $C(\bullet)$  represents the output of discriminator and  $\sigma(\bullet)$  is the sigmoid activation function. And the  $L_G$  equals to the  $L_{adversarial}$  in Equation (3).

Besides, we also apply two-timescale update rule(TTUR) (Heusel et al., 2017) to address slow learning in regularized discriminators. Using regularized discriminator takes more update steps per generator update step during training. TTUR is proposed for discriminator and generator can take fewer update step until convergence, so a faster training is achieved.

## 2.5 Inference

Although the mechanisms mentioned above are applied in our network, they only work in the training process. Therefore, the steganography process won't be affected.

The inference process is straightforward. Given the cover and secret image, we forward them through the encoder and obtain the stego image. The stego image should be saved in the lossless format, such as .png and .bmp. The denoiser can be regarded as a part of the decoder. When the secret image needs to be extracted, we pass the stego image through the decoder and get a clear secret image.

### 3. EXPERIMENT

In this section, we will show our experiment details and the results. In the image-in-image steganography field, ISGAN is an improved version of Baluja (2017) and Rehman (2017)'s work, so only the experiments compared to ISGAN's result is conducted here.

#### 3.1 Experiment Details

All the images fed to the network have been pre-processed to the size of  $256 \times 256$ . The mini-batch size is set to 8. And the learning rate of encode-decoder structure is set to  $1e-4$  at the very beginning and decreases by 0.1 times every 3 epochs. The coefficients in Equation (3) are set as  $\lambda=0.02$  and  $\eta=1$ . While the learning rate of steganalyzer and discriminator is 3 times of baseline. L2 normalization is also adopted during training. As mentioned in Section 2.2, the network is trained in two steps:

- (1) Encoder and decoder are trained with the assistant of steganalyzer until convergence. This process ensures that the encoder can finish the hiding task well. Some experiments performed in Section 3.3 prove our approach can hide images better.
- (2) After the first process, the parameters of encoder are fixed. The stego images to train the decoder and denoiser are in .png format.

For optimization, we use Adam (Kingma & Ba, 2015) optimizer with  $\alpha=0.9$  and  $\beta=0.999$ . The encode-decoder network and discriminator are updated alternately. The hyper parameter setting is all the same in three datasets. Networks are implemented with Pytorch 1.0.1 framework on one NVIDIA GTX 1080Ti.

#### 3.2 Datasets

Our model is trained on subsets of three datasets, Pascal VOC (Everingham et al., 2010), ImageNet (Deng et al., 2009) and MSCOCO (Lin et al., 2014). All of them contain rich and diverse texture. In Pascal VOC 2012 and MSCOCO, 25000 pictures are randomly chosen to constitute 12500 cover-secret pairs to train the model in each dataset, and 7000 pictures are chosen to build 3500 pairs as the test set respectively. As for ImageNet, 50k images from ILSVRC2012 are chosen to constitute 25k pairs to train the model, and 15k images of 7500 pairs are chosen as test set. All the images fed into the network are resized, the shorter margin of which are resized to 256, following a center crop operation.

#### 3.2 Stego Image Quality

In general, image quality is typically evaluated by two widely used distortion metrics, Peak Signal to Noise Ratio (PSNR) and SSIM (Wang et al., 2004a). However, as we all know that PSNR disagrees with the subjective evaluation of human observer (Ledig et al., 2016), so PSNR is regarded as a reference index. The result is shown in Table 1. Our algorithm has improvement in both SSIM and PSNR. The metrics between secret and revealed image is not the final result of our model due to the end-to-end training process, and it only stands for that enough information is embedded in stego image.

As shown in Figure 4, our algorithm outperforms ISGAN visually in some details. Due to perceptual loss, the steganalyzer not only makes the score between stego image and cover image closer, but also their feature maps, both deep features and shallow features. Therefore, the more similar their feature maps are, the more similar their images are. That's why the quality of our stego images is better.

#### 3.3 Security

Since there is no standard metrics for security test, we design an acceptable method to test the security. First, the well-trained models of ISGAN and our algorithm are used to generate 10,000 stego images separately. Then we make two new datasets consist of the stego images generated from two algorithms and their corresponding cover images. As a classifier, a XuNet is trained for 100 epochs to distinguish

Table 1. Image quality in encoding part of our method against ISGAN. SSIM and PSNR index between cover image and stego image of our method are higher than ISGAN. Due to the revealed image in encoding part isn't the final revealed image of our model, the SSIM and PSNR index between revealed image and secret image is for reference.

| Model      | Dataset  | Cover-stego<br>SSIM | Secret-revealed<br>SSIM | Cover-stego<br>PSNR | Secret-revealed<br>PSNR |
|------------|----------|---------------------|-------------------------|---------------------|-------------------------|
| ISGAN      | ImageNet | 0.9558              | 0.9559                  | 34.76               | 33.40                   |
| Our Method | ImageNet | <b>0.9646</b>       | 0.9531                  | <b>37.52</b>        | 35.42                   |
| ISGAN      | VOC 2012 | 0.9523              | 0.9250                  | 34.36               | 33.23                   |
| Our Method | VOC 2012 | <b>0.9592</b>       | 0.9401                  | <b>36.24</b>        | 33.62                   |
| ISGAN      | MSCOCO   | 0.9593              | 0.9366                  | 34.61               | 33.53                   |
| Our Method | MSCOCO   | <b>0.9644</b>       | 0.9360                  | <b>35.38</b>        | 34.32                   |

between cover and stego images for two algorithms. This experiment is conducted on all three datasets separately and the corresponding detection rate is shown in Table 2. The detection rate is defined as how many stego images are detected by the specified XuNet classifier. The lower detection rate means better performance in safety. Obviously, the detection rate of our algorithm is about 10% lower than that of ISGAN. It proves that our modification with perceptual loss does benefit to the security.

Figure 4. Our stego image compared to ISGAN's. (a)ISGAN's result (b)Cover image (c)Our result. Sometimes, ISGAN produce artifacts can be detected by eyes. Note that some grids appear on the background in the first image.

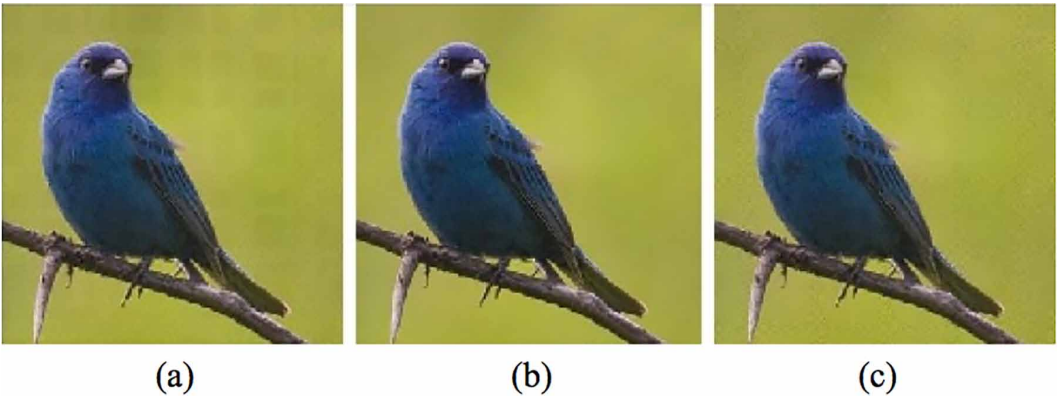
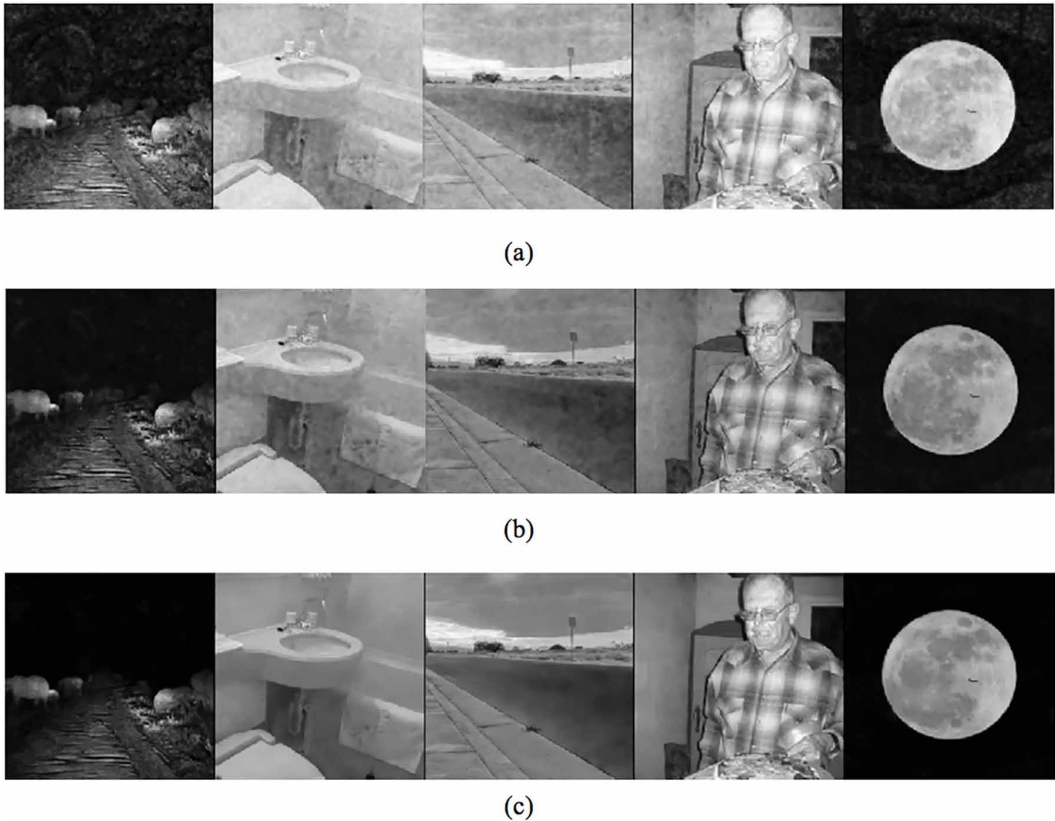


Table 2. Detection rate.

| Model      | ImageNet    | VOC2012     | MSCOCO      |
|------------|-------------|-------------|-------------|
| ISGAN      | 0.84        | 0.85        | 0.84        |
| Our method | <b>0.75</b> | <b>0.75</b> | <b>0.78</b> |



Figure 5. Visual result for different method. (a)Result without finetuned. (b)Finetuned result. (c)Finetuned and denoised result. We take the result from MSCOCO as example. The first row is the result from the end-to-end trained decoder. The second row is the result from finetuned decoder. And the last row is the result after denoiser.



### 3.4 Two Step Training

As mentioned above, the end-to-end network can not be used in practice. Therefore, it's necessary to divide the encoder and decoder into two parts. First, we train our model end-to-end as Baluja (2017) did in their works. The well-trained model is used to generate stego images which must be quantized and saved as a real picture in the hard disk instead of the tensor in the GPU. It is worth mentioning that, we need to save the stego images by lossless method, such as .png and .bmp format, or it will lose too much information. In the lossless saving process, images lose 24 bits information for quantization, from 32 bits to 8 bits, which makes it difficult for decoder to reveal secret images well. Samples on how different decoding methods affect visual quality of revealed image are shown in Figure 5.

### 3.5 Denoiser

Although the finetune process improves the image quality, there still exists severe noise in the revealed image. We append a denoiser to the end of decoder, aiming to remove the noise. In initializing process, the denoiser uses the Xavier initialization while the decoder loads the parameters finetuned in the last step. Unlike decoder, it's a training process for denoiser, so we set the learning rate to 1e-4. While the learning rate is set to 1e-5 for decoder, for which it's a finetuned process. Quantitative results are shown in Table 3 and visual examples are provided in Figure 5. The denoiser successfully removes noise from revealed image and significantly improves the quality of revealed image.

### 3.6 GAN Structure

GAN structure is not only used in the encoder, but also applied to the decoding process to further relieve the effect of noise. The GAN structure helps the decoder learn the mapping from the revealed images to secret images. It encourages the revealed images to move toward the secret images manifold. We evaluate the performance of decoding part with and without adversarial component and the result is shown in Table 3. Following SRGAN (Ledig et al., 2016), although the model without GAN structure, has higher SSIM or PSNR values, however, the results are rather smooth and less convincing perceptually. While the one with GAN performs better. That's caused by the competition between the content loss and the adversarial loss. The adversarial mechanism helps the model produce realistic images. As shown in Figure 6, the result of the model without GAN structure has artifacts in the left-side background and the hair of the monkey, while the one with GAN has better performance. In contrast, the model with GAN structure performs better in the details.

Table 3. Quantitative result for different decoding operation. We take the result from MSCOCO as example.

| Method             | SSIM          | PSNR         |
|--------------------|---------------|--------------|
| End-to-end decoder | 0.8261        | 20.41        |
| Finetuned decoder  | 0.8840        | 26.32        |
| With denoiser      | 0.9059        | <b>27.89</b> |
| Denoiser+GAN       | <b>0.9061</b> | 27.62        |

## 4. CONCLUSION

We have presented a steganography model to hide image in image, which achieves state-of-the-art performance on image steganography. Compared to previous work, two-step training method is proposed, to rectify the shortcoming of end-to-end trained steganography network. Moreover, perceptual loss is added between features to enhance the security and image quality of stego image. In addition, to improve the revealed image quality, we use the adversarial structure in image decoding part and append a denoising structure to the end of the decoder. To stabilize the adversarial training process, useful techniques, like spectral normalization, are adopted.

## REFERENCES

- Abadi M., Andersen D. G. (2016). *Learning to protect communications with adversarial neural cryptography*. Retrieved from <https://arxiv.org/abs/1610.06918>
- Arjovsky, M., Chintala, S., & Bottou, L. (2017, Aug). Wasserstein generative adversarial networks. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70, pp. 214–223). Sydney, Australia: PMLR.
- Baluja, S. (2017). Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 2069–2079). Curran Associates, Inc.
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007, August). Image denoising by sparse 3-d transform domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8), 2080–2095. doi:10.1109/TIP.2007.901238 PMID:17688213
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). doi:10.1109/CVPR.2009.5206848
- Dong, S., Zhang, R., & Liu, J. (2018). *Invisible steganography via generative adversarial network*. Retrieved from <https://arxiv.org/abs/1807.08571>
- Everingham, M., Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010, June). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2), 303–338. doi:10.1007/s11263-009-0275-4
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (Vol. 27, pp. 2672–2680). Curran Associates, Inc.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). *Improved training of wasserstein gans*. Retrieved from <https://arxiv.org/abs/1704.00028>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). IEEE.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., & Hochreiter, S. (2017). *Gans trained by a two time-scale update rule converge to a Nash equilibrium*. Retrieved from <https://arxiv.org/abs/1706.08500>
- Holub, V., & Fridrich, J. (2012, Dec). Designing steganographic distortion using directional filters. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)* (p. 234–239). doi:10.1109/WIFS.2012.6412655
- Holub, V., Fridrich, J., & Denemark, T. (2014, January). Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1), 1. doi:10.1186/1687-417X-2014-1
- Jain, V., & Seung, S. (2009). Natural image denoising with convolutional networks. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems* (Vol. 21, pp. 769–776). Curran Associates, Inc.
- Johnson, J., Alahi, A., & Li, F. (2016). *Perceptual losses for real-time style transfer and super-resolution*. Retrieved from <https://arxiv.org/abs/1603.08155>
- Jolicœur-Martineau, A. (2018). *The relativistic discriminator: a key element missing from standard GAN*. Retrieved from <https://arxiv.org/abs/1807.00734>
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, USD, May 7-9, 2015, conference track proceedings*.
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A. P., Tejani, A., . . . Shi, W. (2016). *Photo-realistic single image super-resolution using a generative adversarial network*. Retrieved from <https://arxiv.org/abs/1609.04802>
- Li, B., Wang, M., Huang, J., & Li, X. (2014, Oct). A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 4206–4210). doi:10.1109/ICIP.2014.7025854

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., & Zitnick, C. L. et al. (2014). Microsoft COCO: common objects in context. In *European Conference on Computer Vision 2014* (pp.740–755). doi:10.1007/978-3-319-10602-1\_48

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). *Spectral normalization for generative adversarial networks*. Retrieved from <https://arxiv.org/abs/1802.05957>

Pevný, T., Filler, T., & Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme, P. W. L. Fong, & R. Safavi-Naini (Eds.), *Information Hiding* (pp. 161–177). Berlin: Springer Berlin Heidelberg. doi:10.1007/978-3-642-16435-4\_13

Rehman, A., Rahim, R., Nadeem, M. S., & Hussain, S. (2017). *End-to-end trained CNN encode-decoder networks for image steganography*. Retrieved from <https://arxiv.org/abs/1711.07201>

Shi, H., Dong, J., Wang, W., Qian, Y., & Zhang, X. (2017). *SSGAN: secure steganography based on generative adversarial networks*. Retrieved from <https://arxiv.org/abs/1707.01613>

Volkhonskiy, D., Nazarov, I., Borisenko, B., & Burnaev, E. (2017). *Steganographic generative adversarial networks*. Retrieved from <https://arxiv.org/abs/1703.05502>

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., . . . Loy, C. C. (2019). Esrgan: Enhanced super-resolution generative adversarial networks. In L. Leal-Taixé & S. Roth (Eds.), *Computer Vision – ECCV 2018 Workshops* (pp. 63–79). Cham: Springer International Publishing.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004a). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 600–612. doi:10.1109/TIP.2003.819861 PMID:15376593

Wang, Z., Simoncelli, E., & Bovik, A. (2004b). Multiscale structural similarity for image quality assessment. *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, 2, 1398 - 1402.

Wolfgang, R. B., & Delp, E. J. (1996). A watermark for digital images. In *Proceedings of 3<sup>rd</sup> IEEE International Conference on Image Processing* (Vol. 3, pp. 219-222). doi:10.1109/ICIP.1996.560423

Xu, G., Wu, H., & Shi, Y. (2016). Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5), 708–712. doi:10.1109/LSP.2016.2548421

Yang, J., Liu, K., Kang, X., Wong, E. K., & Shi, Y. (2018). *Spatial image steganography based on generative adversarial network*. Retrieved from <https://arxiv.org/abs/1804.07939>

Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2016). *Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising*. Retrieved from <https://arxiv.org/abs/1608.03981>

*Guanzhong Wu received the BS degrees from South China University of Technology in 2019. Now he is a master student in the School of Electronic and Information Engineering, South China University of Technology. He is researching in image steganography and object detection.*

*Xiangyu Yu was born in Wuhan, China, in 1978. He received the B.S., M.S., and Ph.D. degrees from Wuhan University, Wuhan, China, in 2000, 2002, and 2006, respectively, all in communication engineering. He was a Lecturer with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, from 2006 to 2013, where he has been an Associate Professor with the School of Electronic and Information Engineering, since 2013. He has authored one book and over 20 journal publications, edited for four books. His current research interests include wireless sensor networks, image processing, and signal processing.*

*Hui Liang received the BS and MS degrees from South China University of Technology in 2016 and 2019, respectively. His research interests are image steganography, image steganalysis and deep learning.*

*Minting Li received the MS degrees from South China University of Technology in 2019. Her research interest is deep learning.*