Steganography with Convincing Normal Image from A Joint Generative Adversarial Framework

Hanqi Zi[1], Qiong Zhang[1,*], Jianhua Yang[1] and Xiangui Kang[1,*]
[1]Guangdong Key Laboratory of Information Security Technology, School of Data and Computer Science,
Sun Yat-sen University, Guangzhou 510006, China. *E-mail: zhangq39@mail.sysu.edu.cn; isskxg@mail.sysu.edu.cn.

***Abstract*— Image steganography conceals secret message into digital image without influencing human perception. Recently, a steganographic method based on generative adversarial networks (GANs) has been tentatively applied to human face dataset in order that the generated images can evade being detected by steganalytic methods. In this paper, we propose a more effective GAN-based steganographic framework, named VAE-SGAN, which combines together several deep learning based network structures: the encoder, the decoder/generator, the discriminator, and the steganalyser. This proposed model can generate better visually convincing images with less model collapse. Through comparative experiments, it has been proved that the generated images are more secure against steganalysis than those generated by the previously established GAN-based methods when working under some popular steganography schemes, such as LSB-matching, WOW and S-UNIWARD.**

I. INTRODUCTION

Steganography [1] is a technique concealing a secret message in a type of medium, so that the presence of the hidden message can avoid to be detected by steganalytic methods. In the fast-growing internet, there has been an abundance of images, audios and videos, of which the security urgently needs to be better guaranteed in covert communications, thereby the design of a secure stegnographic scheme is of much importance.

However, how to develop an effective and secure steganographic framework is a tricky issue that researchers have always concerned about and pursued to solve, while the existing approaches [2] usually engage in the aspect of embedding means, such as falsifying image pixel bits, designing distortion functions, etc.

Least Significant Bit (LSB) matching [3] is one of the most popular embedding schemes in spatial-domain steganography. This algorithm randomly adds or subtracts 1, called ±1-embedding, into the least significant bits of pixel values of an image, but it might deteriorate statistical characteristics of the original image and so that makes the stego image easy to be detected.

Except the LSB method, there exist other effective schemes employing a distortion function to select the embedding localization of the image, which are considered as content-adaptive steganography, and these popular schemes can achieve good security in spatial domain. The distortion function was delicately designed in classic spatial-domain approaches such as the highly undetectable stego (HUGO) [4], wavelet obtained weights (WOW) [5], spatial universal wavelet relative distortion (S-UNIWARD) [6], minimizing the power of optimal detector (MiPOD) [7], and high-pass low-pass low-pass (HILL) [8].

It has been recently reported [9-12] that the generative adversarial networks (GANs) architecture has shown great advantage in generating sensible images by minimizing the differences of data distribution between samples and cover data. The original GAN model was proposed by Ian Goodfellow *et al*. [9] in 2014, but it was proven to be unstable in training and often produce unrealistic images. An obvious disadvantage of the framework is that any differentiable function is theoretically permitted in the model design, which makes the computation process difficult to be controlled in the case of training large images with lots of pixels.

Deep Convolutional GAN (DCGAN) [10] is another GAN-based scheme which integrates supervised learning (from CNN) into unsupervised learning (from GAN), where the generator and discriminator networks can learn hierarchical representations of input images separately. DCGAN is well designed in the aspect of topology and thus is more stable, which has been applied to different tasks of generating images. But the training of DCGAN model will sometimes collapse into an oscillating mode and thereby produce implausible images.

Volkhonskiy *et al*. first combined steganography with the mechanism of GAN and proposed steganographic generative adversarial networks (SGANs) [11]. In their framework, except for the modules of discriminator and generator as similar as in the DCGAN model, another adversarial network called steganalyser was also introduced. The authors used CelebA dataset [12] to train the SGAN model and generate images using different random noise seed values as input. SGAN model shows the capability of acting as a container for steganographic applications. However, the steganalyser they employed in the model is so simple that the generated cover images produced by SGAN are not secure enough for steganography tasks and are prone to be implausible in human visual sense.

Recently, Hu *et al*. [13] conducted to establish formal connections between GANs and VAEs (Variational Auto-Encoders), and they revealed the relationships between these two network structures in a manner of unified interpretation, which theoretically explained the feasibility of the combination of generative models. Besides, many other reports [14,15,16] have also proved that strengthened VAEs can be directly applied to GANs to improve the model quality, whereas the improved GANs can also be added into VAEs to achieve a boost of performance.

In this paper, based on the fundamentals of generative

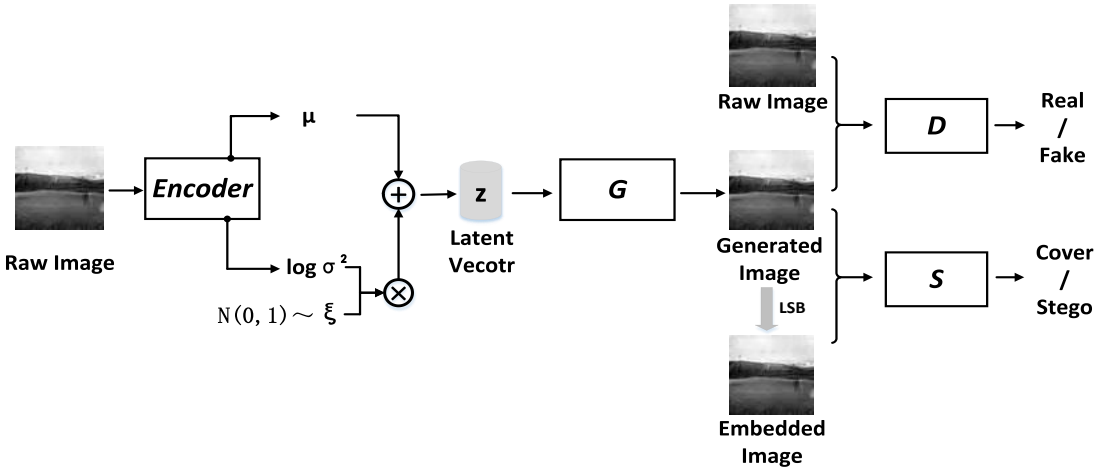


Fig. 1. Architecture of the proposed VAE-SGAN framework.

adversarial networks and variational auto-encoders, we propose a steganography-oriented framework, named VAE-SGAN, to generate convincing normal images more effectively and efficiently, in order that much better security can be achieved in spatial image steganography.

II. The Proposed Framework

A. Overall Architecture

Our proposed deep learning structure, named VAE-SGAN as mentioned above, is made up of four neural networks, which are trained synergistically to make the net generate secure cover images other than merely visually convincing images.

The detailed architecture can be shown as in Fig. 1, where the encoder receives raw images and characterizes features by outputting low-dimensional latent vectors. The decoder, also acting as the generator *G*, receives 128-dimensional latent vectors through iterative training in VAE model to make generated images revised gradually close to raw images. The discriminator *D*, as the opponent of *G*, tries to detect images as raw or generated, which guarantees the visual quality of generated images. Meanwhile, the steganalyser *S* tries to distinguish the stego images from the generated cover images. Taken into consideration the efficiency and performance of the implementation of steganalysis, we employ the model presented in [17] (referred as XuNet) as being the steganalyser *S*.

B. Encoder

The encoder network *E* maps the raw image $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$ through a learned distribution $P(\boldsymbol{z}|\boldsymbol{x})$.

We use raw images as input data of the encoder. The encoder *E* contains four 5×5 convolutional layers with 2×2 strides, while each layer is followed by BN (batch normalization) and LeakyReLU (leaky rectified linear unit) activation, where the latter one has been proved to have better performance than ReLU (rectified linear unit) activation in convolutional neural networks [18]. The output of the last convolutional block is flattened and reshaped to be fed into two independent 128-neuron fully-connected (FC) layers respectively. One FC-layer represents the mean of the compressed image, and the other one represents the logarithm of the image variance. We introduce a 128-dimensional vector ξ with standard normal distribution, which can be regarded as a noise to compel the network to generate varieties of plausible images. The latent vector is the summation of the mean vector and the inner product from logarithmic variance and vector ξ. The network structure can

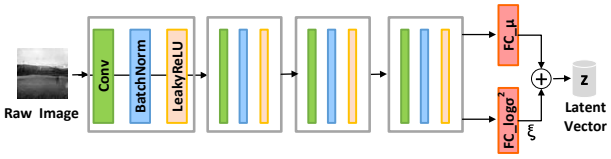


Fig. 2. Architecture of the encoder.

be described as in Fig. 2.

C. Decoder/Generator

As is known, the generator *G* is used to generate secure cover images.

In the generator, the 128-dimensional latent vector *z* is input into a series of blocks utilizing the fractionally-strided convolution, which is referred as FS-Conv for short. Inspired by [10], we use a fully-connected layer with 8192 neurons to be the input layer, and then reshape it into a 4-dimensional tensor, revising the matrix multiplication with BN followed by ReLU activation, as being the start of the convolution stack. Afterwards, four blocks are concatenated, while each of the first three blocks starts with a FS-Conv layer with 5×5 kernels and 2×2 strides, followed by BN, and ends with ReLU activation. Finally, a hyperbolic tangent function (TanH) layer

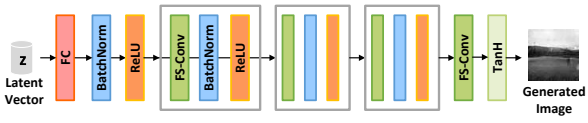


Fig. 3. Architecture of the decoder.

is straightly connected to the last FS-Conv layer to be the output layer. This network structure can be shown as in Fig. 3.

D. Discriminator

To evaluate the human visual quality of the generated images produced by *G*, a discriminator *D* is also designed.

Here, we do not adopt a complex model for *D* to prevent it to be over strong, otherwise *G* might be restrained too much to move on in iterations of the computation. The discriminator *D* utilizes the generic strided convolution, which guarantees the visual quality of the generated images. We use both of raw images and generated images as input data. The discriminator *D* contains four 5×5 convolutional layers with 2×2 strides, and each Conv layer is followed by BN and LeakyReLU. It is proven that using LeakyReLU in GAN framework is beneficial for the stability of the training process [19]. The output of the last convolutional layer is flattened and reshaped to be fed into a fully-connected layer with only one neuron. This network structure can be described as in Fig. 4.

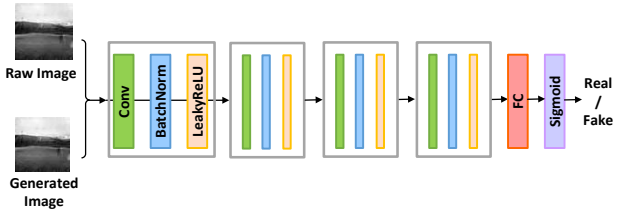


Fig. 4. Architecture of the discriminator.

E. Steganalyser

In regard to network *S*, it is designed to evaluate the security of a stego image using a generated image as being cover.

As mentioned before, we introduce XuNet [17] to form the steganalyser taking into account the quality and performance of steganalytic implementation, making the generated images catered to steganographic tasks. It is worth noting that XuNet employs KV (Ker-Bohme) high-pass filter [20] at the beginning, which suppresses image content and preserves high-frequency part to obtain a steganalysis-oriented noise residual. Inside the first convolutional block, an absolute activation (ABS) layer is inserted to force the statistical modeling to take into account the (Sign) symmetry [21] existed in noise residuals. The structure of the steganalyser network is demonstrated as in Fig. 5.

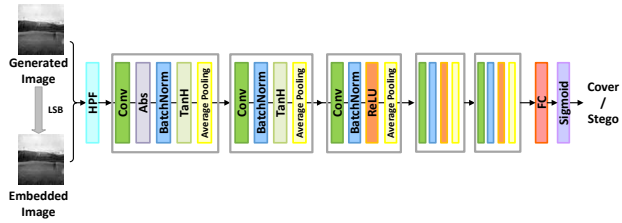


Fig. 5. Architecture of the steganalyser.

III. Design of Loss Functions

Most of the loss functions referred in our proposed network are designed based on cross entropy, which is a common measure in deep learning for the similarity between two probability distributions. The total loss of our framework is comprised of four parts.

A. Loss of Encoder

E is a network which encodes a data sample into a latent representation, so that low dimensional features can be better captured. Conventional GAN-based models usually employ random noise as input signal; on the contrary, we use latent vector as an incentive to the generator, which produces a more sensible signal for the subsequent training process. The loss function of generator *G* can be modeled as follows [22]:

$$L_E = \gamma L_{KL} + \delta L_{rec} \tag{1}$$

where we choose $\gamma = 0.025$ and $\delta = 1$ in the experiments. The settings of γ and δ are based on the magnitudes of L_{KL} (loss of KL divergence) and L_{rec} (loss of reconstruction), respectively.

In Eq. (1), we have

$$L_{KL} = -0.5 \times (1 + log\sigma^2 - \mu^2 - \sigma^2) \tag{2}$$

where μ and σ^2 represent the mean and variance of the latent vector respectively. L_{rec} is used to measure the differences between raw image and generated image:

$$L_{rec} = \sum_{i=1}^{m} \left(\boldsymbol{x}^{(i)} - G(\boldsymbol{z}^{(i)})\right)^2 \tag{3}$$

where $\boldsymbol{x}^{(i)}$ denotes a natural image and $G(\boldsymbol{z}^{(i)})$ represents a generated sample image aroused by latent vector $\boldsymbol{z}$.

B. Loss of Decoder/Generator

Our ultimate goal is to train a steady and strong generator *G*, which can produce secure images in convincing visual sense for steganographic tasks. As mentioned in Section 2, the encoder is to map data into a low-dimensional vector and the decoder tries to reconstruct the data back, the discriminator *D* is used to guarantee the visual quality of generated images, and the steganalyser *S* is used to evaluate the security of stego images. In our framework, the generator is expected to produce secure cover images other than to merely reconstruct raw images.

To better generate visually convincing images, we use the reconstruction feedback in the loss of *G*, denoted as L_{rec} in Eq. (3), which can regulate image samples and make the samples more reasonable, to measure the gap between raw images and generated images.

From the perspective of generator, we wish the images generated by *G* can deceive the discriminator *D* and meanwhile *D* is difficult to discriminate generated images from original cover images as much as possible. Thus, setting the output of *D* as logits, fed with generated images, we have L_G^D (loss from discriminator) as:

$$L_G^D = -\sum_{i=1}^{m} \log D\left(G(\boldsymbol{z}^{(i)})\right) \tag{4}$$

With similar mechanism, we wish the steganalyser *S* more unlikely to distinguish stego images from cover images in the view of generator. Setting the output of *S* as logits, fed with embedded generated images, we have L_S^D (loss from steganalyser) as:

$$L_G^S = -\sum_{i=1}^{m} \log(1 - S(\text{Stego}(G(\boldsymbol{z}^{(i)})))) \quad (5)$$

In functions (4) and (5), $G(\boldsymbol{z}^{(i)})$ represents a synthetic image for the input latent vector $\boldsymbol{z}^{(i)}$, $D(\boldsymbol{x})$ denotes the output of D fed with *x* as input, $S(\boldsymbol{x})$ indicates the output of *S* fed with *x* as input, $\text{Stego}(\boldsymbol{x})$ stands for the result of embedding some hidden message in the cover $\boldsymbol{x}$. These two equations used as loss functions instead of the ones used in [11] can save computation time and accelerate convergence without influencing the training performance.

Considering the tradeoff of the feedbacks from *E*, *D* and *S*, the loss function of generator *G* can be modeled as follows:

$$L_G = \eta L_{rec} + \alpha L_G^D + \beta L_G^S \quad (6)$$

where we choose η=1, α = 0.1 and β = 0.2 in the experiments. The settings of η, α and β are based on the magnitudes of L_{rec}, L_G^D, and L_G^S, respectively.

C. Loss of Discriminator

The network of *D* is a GAN-oriented structure, which is similarly inverse to the structure of *G*. *G* and *D* contend against each other and can directly obtain the feedback from the rival to update the training to achieve better performance. But *D* needs to classify whether an image is real or generative. The discriminator has to be trained with the loss function as:

$$L_D = -\sum_{i=1}^{m} \log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \quad (7)$$

D. Loss of Steganalyser

We employ XuNet [17] as being the steganalyser *S* to ensure the training efficiency and performance. *S* determines whether an image contains a covert message, and it is another opponent that *G* needs to compete against. With similar methodology, the loss function of *S* is defined as:

$$L_S = -\sum_{i=1}^{m} logS\left(Stego\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) + log\left(1 - S\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \quad (8)$$

IV. Experiments

A. Experimental Setup

The proposed networks were implemented by using TensorFlow framework [22] and were trained on a machine equipped with NVIDIA GeForce GTX-1060 GPU.

Firstly, we use images from BOSSbase v1.01 [23] as the input of *E* to train VAE-SGAN. In order to make *G* simulate the distribution of raw images and generate steganography-oriented secure covers, we also use the same dataset as one of the inputs of *D*. Besides, to make VAE-SGAN model better simulate the distribution of the raw data, we do not crop images at all.

Before the training process, all the weight matrices of the encoder *E*, generator *G*, discriminator *D* and steganalyser *S* are initialized by Gaussian distributions with μ = 0 and σ = 0.02, and the bias vectors $\boldsymbol{b}$ are initialized to zeros. We set batch size to 64 and train VAE-SGAN for 200 epochs. For each minibatch, *E*'s weights are updated as a first step, and then the output is fed into *G*. Afterwards, the generated images, which are the output of *G*, will be input into *D* and *S*. The weights of *D* and *S* are updated for once after the weights of *E* and *G* are updated for twice.

To update the encoder, we use Adam optimization [24] with a fixed learning rate of 0.0002, and the parameters are updated to be α = 0.5, β = 0.999, and ε =10^{-8}. *G* and *D* share the same optimization rule as *E*. The leak slope of LeakyReLU in *E* and *D* is set to 0.2. The mini-batch gradient descent is used to train *S*, in which the momentum is fixed to 0.9, while the learning rate is initialized to 0.001 and scheduled to decrease by 10% for every 5000 iterations.

Secondly, to evaluate security performance of the generated cover images from VAE-SGAN, SRM [17] and XuNet are utilized as two different steganalysers. The performance of security is quantified by the detection accuracy of a steganography scheme against a given steganalyser. In the experiments for evaluation, we use the same 10000 generated cover images produced by VAE-SGAN. To train XuNet, we randomly divide 10000 cover-stego pairs into three parts: 7000 pairs for training, 1000 pairs for validation, and 2000 pairs for testing. To train SRM, we combine the training set with the validation set, and maintain the same left 2000 pairs for testing.

Thirdly, for comparison, experiments on natural images from the BOSSbase dataset are also conducted, which is utilized to measure the feasibility of VAE-SGAN model. Here, we adopt 64×64 images, with the same size as the generated images. Besides, comparative experiments based on DCGAN model and SGAN model are also conducted to have proved that

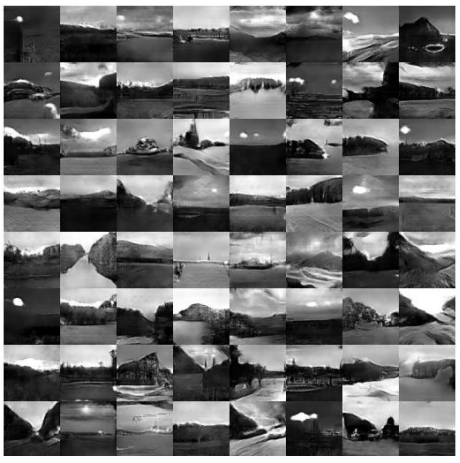
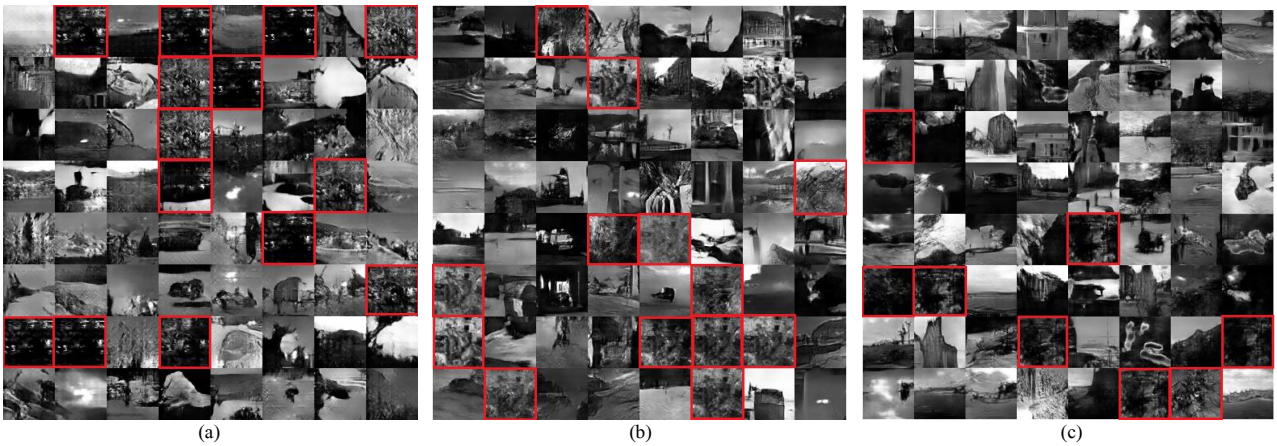


Fig. 6. Generated image samples from VAE-SGAN cover dataset



(a) (b) (c)

Fig. 7. Generated sample images produced by frameworks of DCGAN (a), SGAN (b) and VAE-SGAN (c) after 200 epochs, respectively. The samples within red boxes denote the visually implausible, i.e., model-collapsed images.

our proposed method can generate secure cover images, not merely visually convincing images. Furthermore, to assess the generality of the images produced by the generator, mainstream adaptive spatial steganography methods, such as LSB [3], WOW [5] and S-UNIWARD [6], are employed for the comparison of steganographic performance.

B. Visual Quality

A series of experiments have been conducted to evaluate the visual quality of the generated images and the feasibility of our proposed framework. As is known, CelebA dataset [12] only has the category of human faces, and LSUN dataset [25] mainly shows the category of bedrooms, but the BOSSbase dataset contains multiple categories of objects in images. Therefore, using BOSSbase, there are many different types of objects in the generated images, as shown in Fig. 6, such as the moon, cloud, mountain, lake, island, etc.

After training DCGAN, SGAN and VAE-SGAN models respectively for 200 epochs, 64 images can be randomly generated respectively, where 7(a)-7(c) display the generated sample images. It is observed that some features cannot be represented well enough when using the DCGAN model. And most of the samples generated by SGAN, as shown in Fig. 7(b), are pale-colored and some images are also similar from each other. We use red boxes to point out visually implausible images which look similar to each other. In Fig. 7(a), we can find 14 such failed images in red boxes, which indicates the generation rate of plausible images (quality ratio) by DCGAN is (1 - 14/64) $\times$100% = 78.13%. In Fig. 7(b), there are 13 such image samples, thus the quality ratio from SGAN is (1 - 13/64) $\times$100% = 79.69%; meanwhile the quality ratio of generated images from VAE-SGAN, as seen from Fig. 7(c), is (1 - 8/64) $\times$100% = 87.50%. No matter how to conduct the experiments under same settings and how many images are generated eventually, the quality ratios of generated images from these three models remain similar as mentioned above, which means our VAE-SGAN model can generate plausible images with better visual quality.

We can observe the training process of the three models after different training epochs. For DCGAN model, we find some samples are hard to be trained and vary slightly no matter how many training epochs are experienced, and there exist some unstable evolutions. The samples generated by SGAN model usually seem difficult to be identified. But our proposed VAE-SGAN model can maintain a steady training process and can generate images with fairly good quality and less model collapse. The images generated by VAE-SGAN can clearly represent the characteristics of various objects and seem more sensible and plausible than the images obtained by using DCGAN and SGAN. Owing to the structural fusion of VAE and GAN in a reasonable way, the generator of VAE-SGAN not only receives latent vectors resulting from VAE module, but also accepts the feedback from GAN module. VAE has high sensibility when producing images, but it is lack of diversity, due to the reason of reconstruction from latent vectors. Whereas, GAN method always engages in finding a way to cheat the discriminator as much as possible, in order to create the best image samples, thus the results probably have much better diversity. Anyway, our joint model of VAE and GAN guarantees both sensibility and diversity for the image generation.

C. Performance Evaluation

We adopt SRM and XuNet for the evaluation and comparison of security performance of steganography on cover images respectively derived from BOSSbase itself, DCGAN, SGAN and VAE-SGAN.

Table 1 lists the detection accuracy for different cover image datasets with different steganography methods respectively detected by SRM and XuNet. The proposed VAE-SGAN model performs the best as seen from the experimental results with the lowest detection accuracy. The lower accuracy, the better approach for steganography.

Using LSB, VAE-SGAN model achieves a superiority of about 4% to both SRM and XuNet compared with the natural image database, BOSSbase. SGAN performs the worst when it

Table 1. Detection accuracy for different steganography methods with 0.4bpp payload.

Steganography method	Cover Source	SRM	XuNet
LSB	BOSSbase	89.30%	90.25%
	DCGAN	87.58%	87.05%
	SGAN	90.55%	93.93%
	VAE-SGAN	**85.13%**	**86.70%**
WOW	BOSSbase	62.33%	59.85%
	DCGAN	56.08%	55.70%
	SGAN	54.23%	53.70%
	VAE-SGAN	**52.50%**	**52.60%**
S-UNIWARD	BOSSbase	63.86%	60.45%
	DCGAN	57.59%	57.15%
	SGAN	56.15%	55.95%
	VAE-SGAN	**54.93%**	**54.85%**

is applied to the standard LSB steganography, while DCGAN performs the second best.

When WOW and S-UNIWARD methods are applied to the corresponding cover image source datasets, our VAE-SGAN model still obtains the best results on both SRM and XuNet. With WOW, VAE-SGAN outperforms BOSSbase by about 10% on SRM and about 7% on XuNet. Whereas, using S-UNIWARD, VAE-SGAN outperforms BOSSbase method by about 9% on SRM and about 6% on XuNet.

It is demonstrated that our approach not only is capable of producing images without outputting evident distortions in the view of human visual sense, but also can generate secure cover images for different steganographic tasks in spatial domain. VAE-SGAN may be potentially used as an effective tool to generate universal steganographic carrier images in the future.

V. Conclusion

This paper has introduced a framework of steganography-oriented generative adversarial networks, named VAE-SGAN. The advantages of the proposed approach are as follows:

(1) VAE-SGAN model can generate images with better visual quality and less model collapse.
(2) The generated images from using the scheme are secure covers, which perform significantly better in covert communications than those produced by other existing GAN-based generators.
(3) The proposed method works best not only under LSB-matching steganography scheme but also under the adaptive steganography schemes, such as WOW and S-UNIWARD.

Acknowledgment

This work was supported by NSFC (Grant Nos. U1536204, 61772571, 61702429) and the special funding for basic scientific research of Sun Yat-sen University (Grant No. 6177060230). The authors would like to thank Mr. Jianhua Yang for his meaningful advice on partial design of the network architecture.

References

[1] A. D. Ker et al., “Moving steganography and steganalysis from the laboratory into the real world,” in Proc. 1st ACM Workshop Inf. Hiding Multimedia Secur, 2013, pp. 45–58.
[2] M. S. Subhedar and V. H. Mankar, “Current status and key issues in image steganography: A survey”, Computer Science Review 13-14.C(2014):95-113.
[3] J. Mielikainen, "LSB matching revisited," in IEEE Signal Processing Letters, vol. 13, no. 5, pp. 285-287, May 2006.
[4] T. Pevny, T. Filler, and P. Bas, “Using high-dimensional image models to perform highly undetectable steganography,” in Proc. 12th Information Hiding Workshop, 2010, pp. 161–177.
[5] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, 2012, pp. 234-239.
[6] V. Holub, J. Fridrich and T. Denemark, “Universal distortion function for steganography in an arbitrary domain”, EURASIP Journal on Information Security, vol. 2014, no. 1, 2014.
[7] V. Sedighi, R. Cogranne and J. Fridrich, "Content-Adaptive Steganography by Minimizing Statistical Detectability," in IEEE Transactions on Information Forensics and Security, vol. 11, no. 2, pp. 221-234, Feb. 2016.
[8] B. Li, M. Wang, J. Huang and X. Li, "A new cost function for spatial image steganography," 2014 IEEE International Conference on Image Processing (ICIP), Paris, 2014, pp. 4206-4210.
[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, “Generative adversarial nets,” in Proc. Advances in Neural Information Processing Systems, 2014.
[10] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv: 1511.06434v2 [cs.LG], 19 Nov 2015.
[11] D. Volkhonskiy, I. Nazarov, B. Borisenko and E. Burnaev, “Steganographic Generative Adversarial Networks,” NIPS Workshop on Adversarial Training, Barcelona, 2016.
[12] Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 3730-3738.
[13] Z. Hu, Z. Yang, R. Salakhutdinov, E. Xing , “On Unifying Deep Generative Models,” in Proc. International Conference on Learning Representations, 2018.
[14] J. Bao, D. Chen, F. Wen, H. Li and G. Hua, "CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 2764-2773.
[15] X.Di, V. M. Patel, “Face Synthesis from Visual Attributes via Sketch using Conditional VAEs and GANs,” arXiv: 1801.00077 [cs.CV], January 2018.
[16] J. Kos, I. Fischer and D. Song, "Adversarial Examples for Generative Models," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, 2018, pp. 36-42.
[17] G. Xu, H. Wu and Y. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis," in IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708-712, May 2016.
[18] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” arXiv: 1505.00853 [cs. LG], 5 May 2015.
[19] S. Chintala, E. Denton, M. Arjovsky, and M. Mathieu, “How to train a GAN? Tips and tricks to make GANs work,” accessed 25 July 2017. [Online]. Available: https://github.com/soumith/

[20] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, “JPEG-phase aware convolutional neural network for steganalysis of JPEG images,” in Proc. 5th ACM Workshop on Information Hiding and Multimedia Security, 2017, pp. 75–84.
[21] J. Fridrich and J. Kodovsky, "Rich Models for Steganalysis of Digital Images," in IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 868-882, June 2012.
[22] M. Abadi, P. Barham, J. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “TensorFlow: A system for large-scale machine learning,” arXiv:1605.08695 [cs.DC], 27 May 2016.
[23] P. B. T. Filler and T. Pevny, “Break our steganographic system—The ins and outs of organizing BOSS,” in Proc. 13th Information Hiding Workshop, 2011, pp. 59–70.
[24] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv: 1412.6980 [cs.LG], 22 Dec 2014.
[25] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, J. Xiao, “LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop,” arXiv:1506.03365 [cs.CV], 10 Jun 2015.
[26] D.P Kingma, M. Welling, “Auto-Encoding Variational Bayes,” arXiv: 1312.6114 [cs.LG], 1 May 2014.