This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2017.2745572, IEEE Signal Processing Letters

Automatic steganographic distortion learning using a generative adversarial network

Weixuan Tang, *Student Member, IEEE,* Shunquan Tan*, *Member, IEEE,* Bin Li, *Member, IEEE,* and Jiwu Huang, *Fellow, IEEE*

***Abstract*—Generative adversarial network has shown to effectively generate artificial samples indiscernible from their real counterparts with a united framework of two sub-networks competing against each other. In this letter, we firstly propose an automatic steganographic distortion learning framework using a generative adversarial network, which is composed of a steganographic generative sub-network and a steganalytic discriminative sub-network. Via alternately training these two oppositional sub-networks, our proposed framework can automatically learn embedding change probabilities for every pixel in a given spatial cover image. The learnt embedding change probabilities can then be converted to embedding distortions, which can be adopted in the existing framework of minimal-distortion embedding. Under this framework, the distortion function is directly related to the undetectability against the oppositional evolving steganalyzer. Experimental results show that with adversarial learning, our proposed framework can effectively evolve from nearly naïve random ±1 embedding at the beginning to much more advanced content-adaptive embedding which tries to embed secret bits in textural regions. The security performance is also steadily improved with increasing training iterations.**

***Index Terms*—Generative adversarial network, Steganography, Steganalysis, Embedding Change probabilities**

I. Introduction

DEEP-LEARNING frameworks, especially CNNs (Convolutional Neural Networks) [1], have achieved state-of-the-art performances in many image classification tasks [2], [3]. Researchers have also tried to investigate the potential of deep-learning frameworks in image steganalysis. Tan et al. firstly explored the application of stacked convolutional auto-encoders, a specific form of CNN in image steganalysis [4]. Qian et al. proposed a CNN based steganalyzer which achieves performance close to the state-of-the-art hand-crafted feature set SRM [5], and demonstrated its transfer ability [6]. Xu et al. constructed another CNN-based steganalyzer [7], [8] equipped with BN (Batch Normalization) layers [9], 1×1 convolutions and global pooling operations [10]. It has competitive performance compared with SRM. In [11], Zeng et al. proposed a hybrid deep-learning framework for JPEG steganalysis incorporating the domain knowledge behind steganalytic rich models [5].

However, according to our best knowledge, there still has been no report on the application of deep-learning in image steganography. Recently proposed GAN (Generative Adversarial Networks) [12] inspires a new approach to deep-learning based steganography. GAN aims to generate artificial samples indiscernible from their real counterparts via competition between a generator and a discriminator. It has achieved state-of-the-art performance in many fields, including super-resolution imaging [13], image synthesis [14]–[17], image inpainting [18], and representation learning [14], [19]. As the rivalry between the generator and the discriminator in GAN resembles the competition between steganography and steganalysis, it is straightforward to develop a GAN based steganographic scheme.

In present advanced spatial steganographic schemes, such as HUGO [20], S-UNIWARD [21], MiPOD [22] and HILL [23], the distortion function is heuristically designed and is not directly related to the undetectability [24]. In this letter, we firstly propose an Automatic Steganographic Distortion Learning framework with GAN (ASDL-GAN for short) for spatial cover images, in which the implicit distortion function is directly related to the undetectability against the oppositional evolving steganalyzer. ASDL-GAN simulates the rivalry between steganography with additive distortion and deep-learning based steganalysis in such a way that the steganographic generator produces stego images according to the learnt distortion function, while the steganalytic discriminator classifies between the cover and the produced stegos. By this means, both sides can evolve based on the other's reaction and the implicit distortion function can be automatically learnt.

II. Our proposed ASDL-GAN framework

A. Overall architecture of ASDL-GAN framework

In our work, we follow the most successful steganographic approach in which the payload is embedded while minimizing a suitably defined additive distortion function [25]:

$$D(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{H} \sum_{j=1}^{W} \rho_{i,j} |x_{i,j} - y_{i,j}| \tag{1}$$

where $\mathbf{X} = (x_{i,j})^{H \times W}$ and $\mathbf{Y} = (y_{i,j})^{H \times W}$ denote cover and stego images respectively. H and W are the height and width

This work was supported in part by the NSFC (61772349, U1636202, 61572329), NSF of Guangdong province (2014A030313557), Shenzhen R&D Program (JCYJ20160328144421330), and Faculty Startup Grant of Shenzhen University (2016052). *(Corresponding author: Shunquan Tan.)*

W. Tang is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. S. Tan is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. B. Li and J. Huang are with the College of Information Engineering, Shenzhen University, Shenzhen, China.

All the members are with Shenzhen Key Laboratory of Media Security, Guangdong Province, 518060 China. (e-mail: tansq@szu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2017.2745572, IEEE Signal Processing Letters

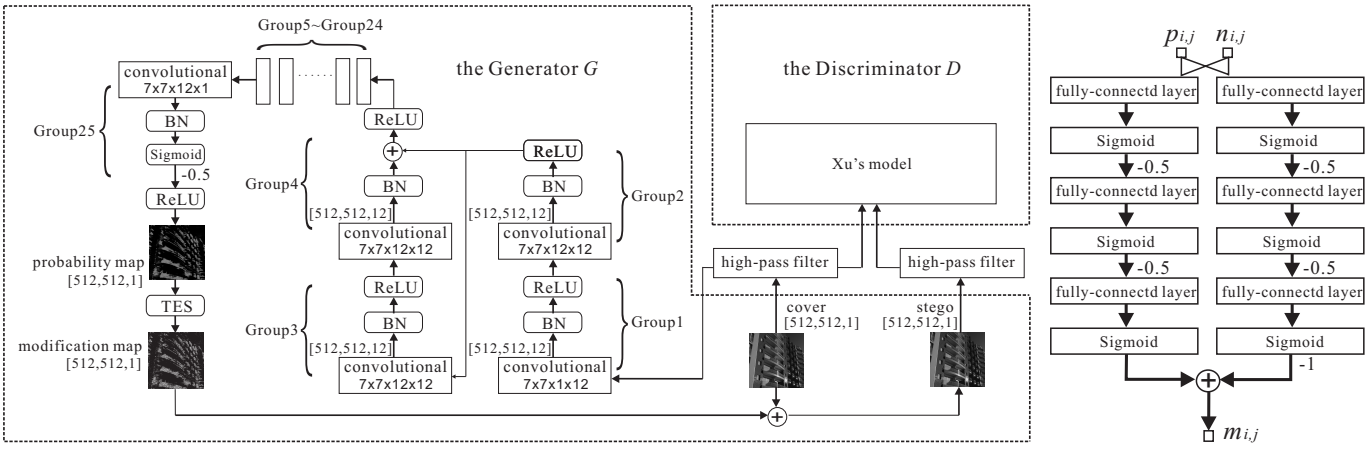

Fig. 1. (a) Architecture of our proposed ASDL-GAN framework. The discriminator D adopts Xu's model [7]. In the generator G, the configurations of the groups from Group2 to Group24 are the same, therefore we use dotted line to Group5 ~ Group24 for simplicity. The kernel configurations of the convolutional layers are given in the following format: (kernel width × kernel height × number of input feature maps × number of output feature maps). The sizes of the output feature maps are given in the format of [height, width, number of feature maps]. (b) Interior structure of the TES activation function.

of an image. $\rho_{i,j}$ is the cost of changing pixel $x_{i,j}$ to $y_{i,j}$. $\mathbf{Q}=(\rho_{i,j})^{H\times W}$ is heuristically designed in a content-adaptive manner in all of the state-of-the-art spatial steganographic schemes, including HUGO [20], S-UNIWARD [21], MiPOD [22] and HILL [23]. For a given embedding payload and the embedding costs $\mathbf{Q}$, the change probabilities $\mathbf{P}=(p_{i,j})^{H\times W}$ can be obtained for every pixel via minimizing the expectation of the distortion function [26]. We take the opposite approach in which the change probabilities $\mathbf{P}$ are learnt firstly and then the corresponding embedding costs $\mathbf{Q}$ can be conversely derived for practical steganographic coding scheme, e.g., Syndrome-Trellis Codes [25], as shown in [22]:

$$\rho_{i,j}=\ln(1/p_{i,j}-2) \tag{2}$$

In our proposed ASDL-GAN framework, the generator G is used to learn the change probabilities $\mathbf{P}$ for every pixel in an input cover image. Its opponent, the discriminator D tries to distinguish the stego images (generated using the change probabilities provided by G) from the innocent cover images. The detailed architecture of the generator G is described in Sect.II-B. Due to considerations of implementation efficiency and steganalytic performance, we employ the model architecture proposed in [7] (referred as Xu's model in this paper) for the discriminator D. In order to generate stego images according to the change probabilities provided by G, we propose a new activation function called TES (Ternary Embedding Simulator) to simulate the process of secret data embedding, which is implemented as a mini neural network and is described in Sect.II-C. The overall architecture of ASDL-GAN is illustrated in Fig. 1.

B. Architecture of the generator G

In all of the previous works [4], [6]–[8], [27], the input images are filtered with a high-pass filter (usually a KV filter), and then the resulting residual maps are fed to the deep-learning steganalyzers. As illustrated in Fig. 1(a), in our proposed ASDL-GAN framework, the discriminator D adopts the model architecture proposed in [7] (referred as Xu's model) and takes the residual maps generated by the KV high-pass filter as input. Therefore the generator G should also take the residual maps generated by the same KV high-pass filter as input to guarantee that the rivalry between the discriminator D and the generator G is on equal footing.

The generator G can be divided into 25 groups. Each of the first 24 groups starts with a convolutional layer with 7×7 convolutional kernels, followed by a BN layer, and ends with a ReLU activation layer. Shortcut connections have been verified to be an effective structure in deep-learning [28]. As a result, in the first 24 groups of layers of the generator G, shortcut connections are added after every two groups to skip the next two groups. The last group outputs a single feature map, in which the element with index (i, j) represents $p_{i,j}$, the change probability of pixel $x_{i,j}$. Sigmoid is used as the activation function on top of the last BN layer so that the output values are within $(0, 1)$. The outputs are further subtracted by 0.5 and then fed in the last ReLU activation function to guarantee that the final outputs of the generator G lie with $(0, 0.5)$. This is due to the fact that modifying pixels with large probabilities may leave potential security holes for steganalyzers, as reported in [29]. Besides, we observed that in state-of-the-art spatial steganographic schemes [21]–[23], almost all change probabilities have values within $(0, 0.5)$. Therefore we take this fact as prior knowledge and force all change probabilities output by the generator G lie within $(0, 0.5)$.

We adopt a ternary embedding scheme, and denote the probability, that a given pixel $x_{i,j}$ is modified to $x_{i,j}+\phi$, as $p^{\phi}_{i,j}$ where $\phi\in\{+1,-1,0\}$. Given a probability map $\mathbf{P}$ outputted by the generator G, we set $p^{+1}_{i,j}=p^{-1}_{i,j}=p_{i,j}/2$ while $p^{0}_{i,j}=1-p_{i,j}$. As a result, the capacity (in bits) of the corresponding stego image can be calculated as follows:

$$\textbf{capacity}=\sum_{i=1}^{H}\sum_{j=1}^{W}(-p^{+1}_{i,j}\log_2 p^{+1}_{i,j}-p^{-1}_{i,j}\log_2 p^{-1}_{i,j}-p^{0}_{i,j}\log_2 p^{0}_{i,j}) \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2017.2745572, IEEE Signal Processing Letters

The output $\mathbf{P}$ of the generator G is further fed in a specific TES activation function (See Sect.II-C). The outputs of the TES activation function constitute a modification map $\mathbf{M} = (m_{i,j})^{H\times W}$, which is added to the corresponding cover image $\mathbf{X}$ to produce stego image $\mathbf{Y}$.

C. Model structure of the TES activation function

When training ASDL-GAN, stego images are constantly generated according to the change probabilities provided by the generator G. In the previous works [20]–[23], a hand-crafted optimal embedding simulator [26] is used to generate stego images. The optimal embedding simulator takes a given change probability $p_{i,j}$ and a random number $n_{i,j}$ from a uniform distribution on the interval $[0, 1]$ as input and outputs the corresponding modification $m'_{i,j}$:

$$m'_{i,j} = \begin{cases} -1, & \text{if } n_{i,j} < p_{i,j}/2, \\ 1, & \text{if } n_{i,j} > 1 - p_{i,j}/2, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

However, as a staircase function, (4) cannot be put in the pipeline of backpropagation in the training procedure of ASDL-GAN. As a gradient-descent friendly alternative, we propose a specific TES (Ternary Embedding Simulator) activation function, which itself is a mini neural network.

Taking $p_{i,j}$ and $n_{i,j}$ as input, our proposed TES activation function is trained to fit the staircase function (4) and generates an approximate output $m_{i,j}$. To achieve this goal, the TES activation function contains 2 independent 4-layer fully-connected mini sub-networks, both of which are fed with the same pair of $p_{i,j}$ and $n_{i,j}$, as shown in Fig. 1(b). Every hidden layer consist of 10 neurons, with each activated by a Sigmoid function and then subtracted by 0.5 for zero-centering. The output of the second mini sub-network is subtracted by 1. With proper settings of the loss function and training parameters (as shown in Sect. III-A), the output of the first mini sub-network would approximately be 1 or 0 while the output of the second mini sub-network would approximately be -1 or 0. By adding the outputs of these two mini sub-networks, TES can approximately produce ternary embedding modifications.

III. The training strategy of ASDL-GAN

A. Pre-training the TES activation function

As stated in Section II-C, the TES serves as an activation function in our ASDL-GAN framework. Therefore the TES should be well pre-trained to fit the staircase function (4) before the training of ASDL-GAN. For this purpose, the loss function of the TES pre-training stage is defined as

$$l_{TES} = \frac{1}{H}\frac{1}{W}\sum_{i=1}^{H}\sum_{j=1}^{W}(m_{i,j} - m'_{i,j})^2 \tag{5}$$

During the pre-training stage, both inputs of the TES model, $n_{i,j}$ and $p_{i,j}$, are independently sampled from a uniform distribution on the interval $[0, 1]$. All of the parameters of the TES model are randomly initialized from a Gaussian distribution with $\mu = 0$ and $\sigma = 1.0$. Stochastic gradient descent is used to optimize the TES model with 0.01 learning rate and 1000 batch size. After 1×10^6 iterations of training, the TES can simulate literally ± 1 modifications in most cases (higher than 90 percent chance). For the rest, the simulation errors are less than ± 0.05 in average. Please note that the weights of the TES remain fixed in the training stage of ASDL-GAN.

B. Training the ASDL-GAN

In the training of ASDL-GAN, the goal for the discriminator D is to distinguish between the cover and the corresponding stego images generated by the generator G and thus the loss function is defined as:

$$l_D = -\sum_{i=1}^{2} y'_i log(y_i) \tag{6}$$

where y_1 and y_2 are the Softmax outputs of the discriminator D, while y'_1 and y'_2 are the corresponding ground truths. As for the generator G, its loss function contains two parts: the adversarial loss l^1_G, defined as (7), makes the produced stego image hard to be detected by the discriminator D; the entropy loss l^2_G, defined as (8), guarantees that the produced stego image can carry approximate q payload of messages. The **capacity** in (8) is defined in (3).

$$l^1_G = -l_D \tag{7}$$

$$l^2_G = (\textbf{capacity} - H \times W \times q)^2 \tag{8}$$

The total loss for the generator G is the weighted average of those two losses as follow:

$$l_G = \alpha \times l^1_G + \beta \times l^2_G \tag{9}$$

where $\alpha = 10^8$ and $\beta = 0.1$. The settings of α and β are based on the magnitude of l^1_G and l^2_G

Before training, all parameters of the generator G and the discriminator D are initialized from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.01$. Then when training ASDL-GAN, the generator G and the discriminator D are alternately trained in each iteration as follow. Firstly we fed a minibatch of 20 images, including 10 cover images and the corresponding 10 stego images produced by the generator G, into the discriminator D and minimize l_D using mini-batch stochastic gradient descent with 0.01 learning rate. And secondly we fed the same 10 cover images to the generator G and minimize l_G using mini-batch stochastic gradient descent with 1×10^{-10} learning rate. In this training stage, 40,000 512×512 grayscale cover images collected by our laboratory (referred as SZUBase dataset) are used. They are converted from the corresponding full-resolution raw images (in Canon® .CR2 format) using the same script that was used for creating the BOSSBase v1.01 [30] dataset.

IV. Experimental results

A. Experiment Setups

After a certain number of training iterations, we used the generator G in ASDL-GAN to generate stego images for 10,000 BOSSbase v1.01 [30] 512×512 cover images. Two steganalyzers, including deep-learning based Xu's model [7]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2017.2745572, IEEE Signal Processing Letters

TABLE I
THE SECURITY PERFORMANCE OF OUR PROPOSED ASDL-GAN SCHEME

Steganographic scheme		0.1 bpp		0.4 bpp	
		SRM	Xu's model	SRM	Xu's model
Our ASDL-GAN scheme with different training iterations	20,000	25.86%	26.92%	13.50%	9.01%
	60,000	27.70%	32.56%	15.44%	14.23%
	100,000	29.35%	36.49%	16.39%	14.52%
	140,000	32.64%	39.72%	16.81%	15.72%
	180,000	33.02%	40.04%	17.40%	16.20%
S-UNIWARD		40.02%	42.53%	20.22%	20.01%

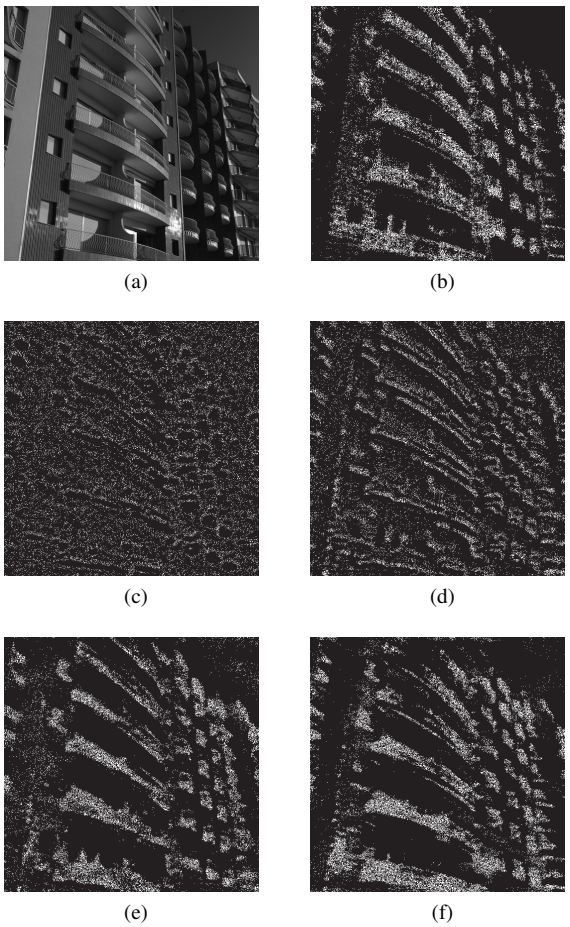

Fig. 2. (a) The BOSSBase cover image "08178.pgm". (b)~(f) Modification maps between the cover image and the corresponding stego images with 0.4 bpp embedding rate, in which (b) is the map generated by S-UNIWARD, (c), (d), (e), and (f) are the maps generated by ASDL-GAN after 500, 5,000, 20,000, and 180,000 iterations, respectively.

and the SRM feature set [5], were used to evaluate the performance of our proposed ASDL-GAN framework. The security performance was quantified by the testing error of a given steganalyzer, which is the average of the false alarm rate and the missed detection rate. As for comparison, experiments on S-UNIWARD [21] were also conducted. Our proposed ASDL-GAN was implemented in TensorFlow. The experiments were conducted on a NVIDIA® Tesla® K80 GPU card.

B. Experimental results and analysis

As defined in (9), the generator G is optimized to approximate the targeted embedding capacity as well as to boost the security performance. In this section, we show the experimental results for these two aspects.

With 0.1 bpp targeted embedding rate, the actual embedding rate averaged over 10,000 BOSSBase images generated by our ASDL-GAN model (trained with 180,000 iterations) is 0.1020 bpp (standard deviation=0.0121). With 0.4 bpp targeted embedding rate, the actual average embedding rate is 0.3885 bpp (standard deviation=0.0235). The above experimental results show that our proposed ASDL-GAN model can well approximate the targeted embedding capacity.

The experimental results of security performance are shown in Tab. I. We can observe that the security performance of our proposed ASDL-GAN is steadily improved with increasing training iterations. To illustrate the evolution of ASDL-GAN along with the increasing iterations, we show the modification maps between BOSSBase image "08178.pgm" and its corresponding stego images generated by ASDL-GAN after different iterations in Fig. 2. From Fig. 2, it can be observed that at the beginning of training, ASDL-GAN did not know which regions were suitable for embedding and therefore the modifications almost randomly spread over the whole scene. With increasing training iterations, ASDL-GAN has learnt to concentrate modifications on textual regions, which are more difficult to model by steganalyzers. Please compare Fig. 2(f) and Fig. 2(b), it is clear that after 180,000 iterations, ASDL-GAN has learnt to embed secret bits in a content-adaptive manner, similar to the modification pattern of S-UNIWARD.

However, we have to acknowledge that even with large training iterations, ASDL-GAN is still inferior to state-of-the-art hand-crafted steganographic algorithms, e.g., S-UNIWARD [21], which may attribute to the lack of training samples. As a large-scale deep-learning framework, ASDL-GAN is hindered by insufficient 40,000 training samples.

V. CONCLUDING REMARKS

The contributions of this paper are as follows.

1) We firstly propose ASDL-GAN, an automatic steganographic distortion learning framework with GAN, which simulates the rivalry between steganography with additive distortion and deep-learning based steganalysis. Under the ASDL-GAN framework, the learnt distortion function is directly related to the undetectability against the oppositional steganalyzer.
2) We implement a steganographic scheme under the ASDL-GAN framework. Experimental results show that the scheme can teach itself to embed secret bits in a content-adaptive manner via alternate training of two oppositional sub-networks.

In our future work, we would like to apply our proposed ASDL-GAN framework to JPEG domain where millions of images are available, and incorporate more advanced deep-learning architectures to boost its security performance.

References

[1] Y. Lecun *et al.*, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015)*, 2015, pp. 1–9.

[4] S. Tan and B. Li, "Stacked convolutional auto-encoders for steganalysis of digital images," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA'2014)*, 2014, pp. 1–4.

[5] J. Fridrich and J. Kodovský, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, 2012.

[6] Y. Qian *et al.*, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proc. IEEE 2016 International Conference on Image Processing, (ICIP'2016)*, 2016, pp. 2752–2756.

[7] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, 2016.

[8] ——, "Ensemble of CNNs for steganalysis: An empirical study," in *Proc. 4th ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2016)*, 2016, pp. 103–107.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[10] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013. [Online]. Available: http://arxiv.org/abs/1312.4400

[11] J. Zeng *et al.*, "Large-scale JPEG steganalysis using hybrid deep-learning framework," *arXiv:1611.03233*, 2016. [Online]. Available: https://arxiv.org/abs/1611.03233

[12] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Advances in Neural Information Processing Systems (NIPS'2014)*, 2014, pp. 2672–2680.

[13] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[15] E. L. Denton *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Proc. Advances in Neural Information Processing Systems (NIPS'2015)*, 2015, pp. 1486–1494.

[16] S. Reed *et al.*, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016. [Online]. Available: http://arxiv.org/abs/1605.05396

[17] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2016. [Online]. Available: http://arxiv.org/abs/1612.03242

[18] D. Pathak *et al.*, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2016)*, 2016, pp. 2536–2544.

[19] X. Chen *et al.*, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," *arXiv:1606.03657*, 2016. [Online]. Available: http://arxiv.org/abs/1606.03657

[20] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. 12th Information Hiding Workshop (IH'2010)*, 2010, pp. 161–177.

[21] 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, pp. 1–13, 2014.

[22] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 221–234, 2016.

[23] B. Li *et al.*, "A new cost function for spatial image steganography," in *Proc. IEEE 2014 International Conference on Image Processing, (ICIP'2014)*, 2014, pp. 4206–4210.

[24] A. Ker. *et al.*, "Moving steganography and steganalysis from the laboratory into the real world," in *Proc. 1st ACM Information Hiding and Multimedia Security Workshop (IH&MMSec'2013)*, 2013, pp. 45–58.

[25] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using Syndrome-Trellis codes," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 920–935, 2011.

[26] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, 2007, pp. 650 502–1–650 502–15.

[27] L. Pibre *et al.*, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch," in *Proc. Media Watermarking, Security, and Forensics, Part of IS&T International Symposium on Electronic Imaging (EI'2016)*, San Francisco, CA, USA, 14-18 February 2016.

[28] K. He *et al.*, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2016)*, 2016, pp. 770–778.

[29] T. Denemark, J. Fridrich, and V. Holub, "Further study on the security of S-UNIWARD," in *Proc. IS&T/SPIE Electronic Imaging 2014 (Media Watermarking, Security, and Forensics)*, 2014, pp. 902 805–902 805.

[30] P. B. T. Filler and T. Pevný, "Break our steganographic system—the ins and outs of organizing BOSS," in *Proc. 13th Information Hiding Workshop (IH'2011)*, 2011, pp. 59–70.