mathematics

MDPI

Article

Enhancing the Security of Deep Learning Steganography via Adversarial Examples

Yueyun Shang [1,2], Shunzhi Jiang [1,*], Dengpan Ye [1,*] and Jiaqing Huang [1]

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430000, China; alice@mail.scuec.edu.cn (Y.S.); JiaqingHuang@whu.edu.cn (J.H.)

[2] School of Mathematics and Statistics of South-Central Universit, Wuhan 430000, China

* Correspondence: jsz@whu.edu.cn (S.J.); yedp@whu.edu.cn (D.Y.)

Received: 23 July 2020; Accepted: 20 August 2020; Published: 28 August 2020

Abstract: Steganography is a collection of techniques for concealing the existence of information by embedding it within a cover. With the development of deep learning, some novel steganography methods have appeared based on the autoencoder or generative adversarial networks. While the deep learning based steganography methods have the advantages of automatic generation and capacity, the security of the algorithm needs to improve. In this paper, we take advantage of the linear behavior of deep learning networks in higher space and propose a novel steganography scheme which enhances the security by adversarial example. The system is trained with different training settings on two datasets. The experiment results show that the proposed scheme could escape from deep learning steganalyzer detection. Besides, the produced stego could extract secret image with less distortion.

Keywords: steganography; information hiding; deep learning; generative adversarial networks; adversarial examples

1. Introduction

Steganography is the science of hiding secret messages in cover images by slightly modifying pixel values that may appear normal to a casual observer. Like cryptography, the steganography technique provides a secret communication method. However, the cryptography method focuses on the authenticity and integrity of the messages. The main goal of the steganography method is to hide the existence of the secret. Massive surveillance operations have shown that even if the content is unknown, the existence of normal data communications may lead to privacy leakage. Therefore, steganography is necessary for private communication.

Steganography techniques could be used in many filed like watermarks, copyright protection, and secret transmission. Usually, the sender uses a steganography algorithm to hide the secret message in the cover, with unaltered to external detectors. The main effort in steganography is to minimize the interference in the cover image when the secret is embedded while allowing the recovery of the secret message Then the steganographic image which is referred to as stego was transmitted in public channels. On the other side, the receiver receives the stego and uses the decoding algorithm and the shared key to extract the secret message.

With the rapid development of deep learning, modifying the image becomes much easier and automatically in steganography. Existing steganography methods could be divided into two categories: STC based content-adaptive steganography and deep learning based automatically steganography. STC(Syndrome-Trellis-CodeI) [1] based content-adaptive steganographic schemes that embed the messages in complex regions are the most traditional secure steganography schemes. Traditional STC

based steganography such as HUGO [2], WOW [3], S-UNIWARD [4], and HILL [5], conceal the secret message into the cover image by modifying the least significant bits of the pixel which imperceptible by human perception or detection tools. The modified pixels are used to encode the secret message. Distinct from traditional STC based steganography method, deep learning based steganography method are learned from machine learning [6–12]. Compared with STC based steganography, deep learning based steganography schemes have higher capacity and the parameters of the steganography algorithm are strongly stochastic.

Corresponding to steganography, steganalysis is a set technique of detecting hidden data in images. For the security of steganography, steganalysis is an important evaluation criterion. Usually, this task is formulated as a binary classification problem to distinguish between cover images and stego images. Compare with tradition SRM (Spatial Rich Model) [13], several deep learning steganalysis methods [14–19] have been proposed to solve the steganalysis problem which improves detection accuracy to a new level.

In this work, we test the current Generative Adversarial Nets based deep learning steganography schemes with Convolution Neural Network (CNN) based deep learning steganalyzers. We found that although the GAN based steganography obtains the algorithm by adversarial training and the structure of discriminator also mostly comes from steganalyzer, the algorithm has poor security in the face of independent steganalyzers. To enhance the security of the embedding algorithm we introduce the idea of adversarial example techniques. Adversarial examples could fool any deep learning classifiers by adding the perturbations produced by backpropagation. It is a general phenomenon in neural networks and just caused by over-fitting or linear behavior in higher-dimensional space. CNN based steganalyzers could also have the same problems. Therefore, the security of steganography could be enhanced by adversarial examples techniques.

We propose a novel steganography scheme that generated the stego image through a novel GAN based model and adversarial example techniques. We show the effectiveness of our scheme in the experiment, the stego produced by encoder could fool deep learning steganalyzers and the extracted secrets are less distorted. The rest of the paper is organized as follows. Section 2 discusses the theory of GAN based steganography and adversarial examples. Section 3 describes the training and enhancing schemes. In Section 4, we show the experiment results and discussions. Finally, conclusions and future works are given in Section 5.

2. Related Work

2.1. GAN Based Steganography

Generative adversarial networks(GANs) [20] have recently led to highly synthesis performance, which widely used in many tasks. Some novel models have been proposed in many image applications, such as style transfer [21], image super resolution [22], image inpainting [23].

Generally, GAN consists of a generator and a discriminator. The task of the generator is to learn to create fake data. The task of the discriminator is to learn to classify the input data is real or fake. The generator and the discriminator are trained simultaneously. At the end of training, the generator can generate high-quality fake data. The generator G and discriminator D play the following two-player minmax game with value function $V(G, D)$:

$$\min_G \max_D V(G, D) = \mathbb{E}_{\boldsymbol{x} \sim p_{data(\boldsymbol{x})}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}(\boldsymbol{z})}}[\log 1 - D(G(\boldsymbol{z}))] \quad (1)$$

Unlike most steganography methods that mainly rely on human handwork, GAN-based steganography algorithms are automatically generated. GAN based steganography methods use an encoder network as G to produce stego and a steganalyzer as D. The encoder and the decoder form the adversarial process in deep learning. The decoder network is used to decode the secret message in the stego image. All the networks are simultaneously trained until the G(encoder) could generate high fidelity stego

images, the decoder could obtain less distorted secret message and D(steganalyzer) could distinguish cover image and stego image with high detection rate.

2.2. Adversarial Examples

Most machine learning classifiers, including deep neural networks, are vulnerable to adversarial examples. This type of input is usually generated by adding small but purposeful modifications that lead to incorrect outputs. Convolution Neural Networks (CNNs) [24] have been widely used in steganalyzer which reach state-of-the-art performance. However, Szegedy [25] pointed out that the function of CNN learning is not continuous, which shows that we can only add a slight perturbation to the original image. Then the image could be misclassified by the classification neural network, and even more, the image could be classified into a specific label.

For a linear model, we have a input $\boldsymbol{x}$ and adversarial input $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\eta}$, $\boldsymbol{\eta}$ is a perturbation. Suppose there is $\|\boldsymbol{\eta}\|_\infty < \epsilon$ which is small enough that will not change the classification of the classifier. Consider the weight w of model, we have:

$$w^T\tilde{\boldsymbol{x}} = w^T\boldsymbol{x} + w^T\boldsymbol{\eta}. \tag{2}$$

The perturbation grows with $w^T\boldsymbol{\eta}$ and activation function. The total perturbation could have a linear growth with the dimensions of w and therefore mislead the model.

The algorithm for generating adversarial examples could be divided into two categories: Gradient based and evolutionary algorithms based. Fast Gradient Sign Method (FGSM) [26] is one of gradient based method which is simple but powerful for image adversarial example. For a deep neural network, the weight is $\boldsymbol{\theta}$, the input image is $\boldsymbol{x}$, the true label of $\boldsymbol{x}$ is y, and the loss is $J(\boldsymbol{\theta}, \boldsymbol{x}, y)$, we have:

$$\boldsymbol{\eta} = \epsilon sign(\bigtriangledown_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)). \tag{3}$$

$\boldsymbol{\eta}$ is the perturbation produced by FGSM. The variation direction of perturbation is consistent with that of the gradient by using $sign$. Then the loss function will increase, which will maximize the change of classification results. The gradient $\bigtriangledown_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ could be obtained from automatic differentiation.

One Pixel Attack [27] is a meta-heuristic optimization algorithm which could fool deep models by changing only one pixel. Let $\boldsymbol{x}$ be the input image, f is the loss function. We have:

$$\arg\max_{\boldsymbol{\eta}} f(\boldsymbol{x} + \boldsymbol{\eta}), \quad where \quad \|\boldsymbol{\eta}\|_0 \leq dim. \tag{4}$$

Only one pixel will be changed if we set dim = 1. Differential Evolution algorithm is used to solve this optimization problem.

3. Steganography Scheme Based on GAN and Adversarial Examples

The security is the crucial point in steganography scheme. For GAN based steganography, the encoder is trained to fool the steganalyzer. However, this does not indicate that the proposed method can counter steganalyzers, because the training strategy in GAN limits the effectiveness of the discriminator. In Section 4, we took the experiment and found that the security of GAN based steganalyzer decrease, when facing independent steganalyzer. We introduce adversarial example techniques to solve the problem. The architecture of our steganography scheme is shown in Figure 1. In this work, our scheme could be divided into two steps: Model training and security improving.

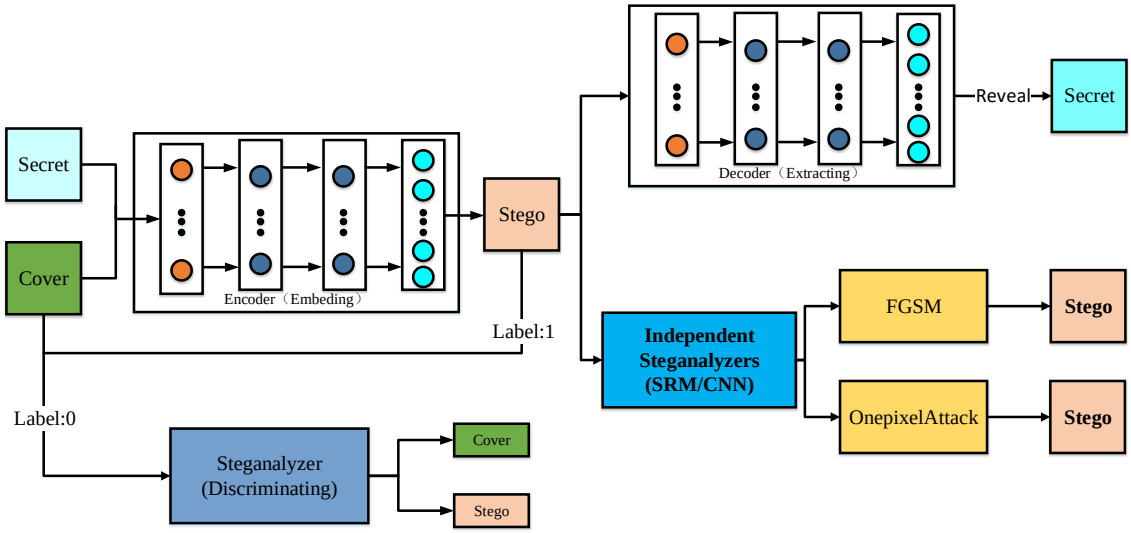

Figure 1. There are two parts in our proposed scheme: The Generative Adversarial Net based steganography module which includes encoder, decoder, discriminator; the security enhancement module which includes independent steganalyzer testing and security stego generating by adversarial example techniques.

3.1. Model Training

In this work, there are three modules in our method: The encoder, the decoder and the steganalyzer. The task of the encoder module is to produce the stego image. The task of steganalyzer is to detect the stego image whether they contain a secret message. The task of the decoder module is to extract the secret message in the stego image. Therefore, we use two deep learning networks to conceal and reveal a secret message. The goal of the model to conceal the secret image within the cover. Thus, the task of the training is discriminative in the model training. First, the encoder module takes in cover images and secret message and outputs stego image. Then, the steganalyzer tries to detect the secret message in the image, resulting in the ability to determine whether the stego contains message.

Denoting $\theta_E, \theta_D, \theta_S$ as the parameters of encoder decoder and steganalyzer. Let A_c, A_i, A_s, A_r for cover, secret, stego and reveal secret. Let $O_e(\theta_E, A_c, A_i)$ as the output of encoder netowrk, $O_d(\theta_D, A_s)$ the output of decoder netowrk, $O_s(\theta_S, A_c, A_s)$ as the output of steganalyzer in the model. We have:

$$O_d(\theta_D, A_s) = O_d(\theta_D, O_e(\theta_E, A_c, A_i)). \tag{5}$$

$$O_s(\theta_S, A_c, A_s) = O_s(\theta_S, A_c, O_e(\theta_E, A_c, A_i)). \tag{6}$$

Let L_e denote the loss of encoder network, L_d denote the loss of decoder network, L_s denote the loss of steganalyzer network. We use the cross entropy between two images as the loss of steganalyzer. The Euclidean distance d between A_i and A_r is used in decoder network reveal loss. The encoder loss is formed by the encoder loss and the steganalyzer loss. λ_a, λ_b, λ_c represent the weight given to each respective loss. Then we have:

$$L_s(\theta_S, A_c, A_s) = -y \cdot log(O_e(\theta_E, x)) - (1-y) \cdot log(1 - O_e(\theta_E, x)). \tag{7}$$

$$L_d(\theta_E, \theta_S, A_c, A_s) = d(A_r, A_i). \tag{8}$$

$$L_e(\theta_E, A_c, A_i) = \lambda_a \cdot d(A_c, A_s) + \lambda_b \cdot L_s + \lambda_c \cdot L_d. \tag{9}$$

3.2. Security Improving

The security is the crucial part in most steganography method. Traditional steganography method change the least significant bits of the pixel of the image, and these changes are difficult to be perceived by humans. However, it could be detected well by the existing steganalysis methods which based on SRM or CNN based model. To verify the security, we use different steganalyzers to detect our steganography method in Section 4. We found that the security of GAN based steganalyzer decrease when facing independent steganalyzer. Considering the actual situation, steganography algorithms usually have to counter independent steganalyers. We introduce two adversarial example techniques to solve the problem.

We use FGSM for white box attack on steganalyzers. Assume that, the details of steganalyzer including parameters, structure, and execution environments are known. Then add the perturbation in the synthesizing stego on gradient direction. To reduce the disturbance between adversarial perturbations and secret, we clip the perturbation in a very small range. One Pixel attack was used for black box attack on steganalyzers. The details of steganalyzer are unknown. While the smaller the change of stego, the better the extraction effect, We do not constrain the number of the changed pixels to 1 for a better attack success ratio.

4. Experiments

In this section, extensive experiments will be conducted to prove the effectiveness of our method. We implemented our scheme with different training settings under two datasets: LFW, Bossbase. The code and experimental data are available under our Github.

4.1. Dataset

The datasets we used in our experiment are The Labeled Faces in the Wild (LFW) [28], Bossbase [29]. The LFW data set contains more than 13,000 face images belonging to 1680 people collected from the web. 10,000 images were Randomly select 10,000 images from LFW and constituted 5000 cover-secret image pairs as our training set, the remaining 3000 face images of LFW were as our validation set. The Bossbase dataset contains 10,000 $512 \times 512 \times 8$ bit grayscale images which have different texture features and are widely used in steganography and steganalysis. Due to the limitations of the graphics memory, we finally evaluate the performance of our scheme on images of 256×256 pixels, which crop the central part of the original images. We randomly selected 4000 cover-secret image pairs as our training set, the remaining 2000 face images of Bossbase were our validation set.

4.2. Implementation Details

We use PyTorch [30] frame to build our model. The input cover and secret images were randomly selected in the training set. We balance between the steganalyzer and decoder losses by using $\lambda_a = 0.5$, $\lambda_b = 1$, $\lambda_b = 1$ on LFW data set. $\lambda_a = 0.8$, $\lambda_b = 1$, $\lambda_c = 1$ for Bossbase data set. We use Adam method to train our model for 150 epochs. The learning rate is set to 0.0001. We selected 1000 attack test samples in every security enhancement Experiments. In FGSM, we use four different ϵ to control the changes to original stegos. In Onepixelattack, we use four different p to control the success rate of the attack, the seed is set to 500, and the maximum number of iterations is set to 50.

4.3. Model Training Experiments

MSE (Mean Square Error) is used to measure the cover image loss. PSNR (Peak Signal to Noise Ration) is used to measure the quality of the stego image and secret image. Table 1 reports the MSE and PSNR for both cover and stego under two validation sets. Note that, the MSE of stegos are less than one percent. Compare with the image in natural environment, the PSNR of the stegos are at a

relatively normal level, which means the trained model could produce high fidelity stego and less distortion secret.

Table 1. MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ration) for both cover loss and secret loss with the GAN based steganography method under two datasets.

	Cover Loss	Secret Loss	Stego PSNR	R-Secret PSNR
LFW	0.0049	0.0061	25.7222	26.3746
Bossbase	0.0175	0.0140	19.8354	21.7831

4.4. Security Enhancement Experiments

During the training process in our GAN based steganography, the encoder was trained to fool the steganalyzer. We found that the model could not counter independent steganalyzer. We created additional 5000 stegos with the trained model. The original covers and the stegos are merged to form a new data set, and then divided into two halves to create a training set and a validation set. We compared our scheme against steganographic methods WOW and S-UNIWARD. For each method, We chose SRM the state-of-art machine learning steganalysis model and a designed CNN based deep learning steganalysis model as the detector. We trained the steganalyzer based on SRM and CNN in the training set, and then report the performance of the steganalyzer on the validation set. We use the detection error (P_E) to compare the performance which using the following form:

$$P_E = \frac{w}{V_n}. \tag{10}$$

V_n is the total number of validation set samples, w is the number of misclassification from the steganalyzer. Table 2 shows the detection errors of independent SRM and CNN based steganalyzer against five steganography methods.

Table 2. The detection errors of independent SRM (Spatial Rich Model) and CNN (Convolution Neural Network) based steganalyzer against five steganography methods.

	LFW		Bossbase	
Algorithms	**SRM**	**CNN**	**SRM**	**CNN**
WOW	0.2587	0.1328	0.2887	0.1654
S-UNIWARD	0.2805	0.1571	0.2704	0.1849
GANste	0.1910	0.1269	0.1039	0.0819
FGSM-GANste $\epsilon = 0.001$	0.1394	0.2147	0.1387	0.1916
FGSM-GANste $\epsilon = 0.003$	0.1704	0.4678	0.1208	0.5576
FGSM-GANste $\epsilon = 0.005$	0.1773	0.7294	0.1135	0.8440
FGSM-GANste $\epsilon = 0.008$	0.1638	0.9423	0.1039	0.9808
Onepixelattack-GANste $p = 1$	0.2202	0.5323	0.1231	0.2265
Onepixelattack-GANste $p = 3$	0.1666	0.3125	0.1190	0.3235
Onepixelattack-GANste $p = 5$	0.2168	0.3333	0.0843	0.1247
Onepixelattack-GANste $p = 5$	0.2667	0.2143	0.1724	0.1615

In the first three rows in Table 2, we compared the security of trained GAN based steganography method with that of two content adaptive steganography methods WOW and S-UNIWARD at 0.4 bpp. Note that, the GAN based steganography method could not escape from detection well when counter the independent steganalyzers that is different from the discriminator. In the last several rows, we used two adversarial example techniques to enhance the security of GAN based steganography. We used four ϵ in FGSM and four p in Onepixelattack, and the corresponding detection error is at a relatively high level when counter deep learning steganalyzer, which indicate the stego could fool the steganalyzers and its security improved. Considering that the state-of-art steganalyers are deep learning based, the adversarial example techniques we used mainly for CNN. Thus, the adversarial

performance is weaker in SRM. Figure 2 shows the confidence prouduced by logits layer under two datasets. Note that, with the increase of ϵ in FGSM, most of the stegos confidences of CNN will decrease which indicates the attack success.

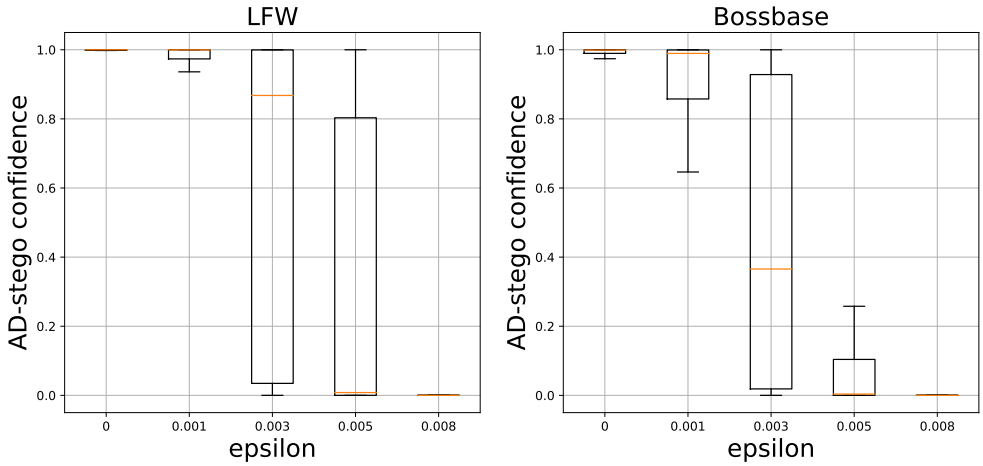

Figure 2. The steganalyzer logits layer confidence comparison between different ϵ in Fast Gradient Sign Method (FGSM) under two datasets.

From the analysis of Table 2 and Figure 2, we find that, when using FGSM, the detection error get higher with the increase of ϵ, which indicate the lager adversarial perturbation, the easier it is to fool the steganalyzer. However, the adversarial is mix with the secert message, strong perturbation could affect the original secret. Thus, we use a relatively low ϵ. Onepixelattck is black box attack which changes one pixel to mislead the classifier. In our dataset, limit to our memory, we chose several pixels to modified. We found that the more piexls modified, the easier fool steganalyzer will succeed. In addition, Onepixelattack causes little distortion. We tested the decoding performance of GAN based steganography after using adversarial example techniques. Table 3 shows the MSE and PSNR results for both modified cover and secret, while Figure 3 shows the visualization of several covers, stegos, secret images and their residuals from two datasets. Note that, the adversarial example techniques could slightly modify the stego produced by encoder but enhance the security while causing little influence on the extraction.

Table 3. MSE and PSNR for both cover loss and secret loss with two training settings under two datasets.

	Cover Loss	**Secret Loss**	**Stego PSNR**	**R-Secret PSNR**
FGSM-LFW $\epsilon = 0.003$	0.0049	0.0079	25.8321	22.4592
FGSM-Bossbase $\epsilon = 0.001$	0.0028	0.0039	27.9957	26.4751
Onepixelattack-LFW	0.0167	0.0148	19.8567	20.0550
Onepixelattack-Bossbase	0.0205	0.0091	20.2960	22.8812

Figure 3. The visualization of several covers, stegos, secret images and their residuals from two datasets. The first and third rows show the embedding performance comparison between GAN based steganography stegos and adversarial example steganography methods. The second and last rows show the original secret image, revealed secret image and the extracting performance comparison between GAN based steganography stegos and adversarial example steganography methods with their residuals.

5. Conclusions

In this paper, we proposed a novel method which enhances the security of deep learning based steganography method. We take advantage of the linear behavior of state-of-art CNN based steganalyzer and use adversarial example techniques to let stego escape from detection. We have proved the effectiveness of our model by implementing different experiments. We also found the single antagonism of the adversarial perturbation, due to the process of generation. We would like to further investigate more universal adversarial example techniques. The distortions of stegos produced by the use of the adversarial technique is evaluated by MSE and PSNR. However, in traditional steganography method, these indicators are not very sensitive to localized distortions, such distortions could lead to very good values of the indicators but on the other hand the distortions could become visible in the stego. Another research direction could further reduce the distortion of adversarial perturbation to secret. We would like to investigate the possibility of looking at post silicon technologies where the security of the information is an open problem; the nanofluidics and microfluidics devices based network that use quite different hardware have the same problems as regards the security of the traditional ones.

Author Contributions: Conceptualization, Y.S. and S.J.; Funding acquisition, D.Y.; methodology, S.J. and Y.S; resources, D.Y.; software, J.H.; formal analysis, Y.S.; writing original draft, Y.S.; writing, review and editing, S.J.; visualization, Y.S.; supervision, D.Y.; project administration, S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded byNational Key Research Development Program of China (2019QY(Y)0206, 2016QY01W0200), and the National Natural Science Foundation of China NSFC (U1736211).

Conflicts of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "Enhancing the Security of Deep Learning Steganography via Adversarial Examples".

References

1. Tomáš, F.; Judas, J.; Fridrich, J. Minimizing embedding impact in steganography using trellis-coded quantization. In *Media Forensics and Security II*; International Society for Optics and Photonics: Bellingham, WA, USA, 2010; Volume 7541.
2. Pevny, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In Proceedings of the International Workshop on Information Hiding, Calgary, AB, Canada, 28–30 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 161–177.
3. Holub, V.; Fridrich, J. Designing steganographic distortion using directional filters. In Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Tenerife, Spain, 2–5 December 2012; pp. 234–239.
4. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**. [CrossRef]
5. Li, B.; Wang, M.; Huang, J.; Li, X. A new cost function for spatial image steganography. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014.
6. Jamie, H.; Danezis, G. Generating steganographic images via adversarial training. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
7. Shumeet, B. Hiding images in plain sight: Deep steganography. In Proceedings of the Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.
8. Zhu, J.; Kaplan, R.; Johnson, J.; Li, F. Hidden: Hiding data with deep networks. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
9. Li, S.; Ye, D.; Jiang, S.; Liu, C.; Niu, X.; Luo, X. Anti-steganalysis for image on convolutional neural networks. *Multimed. Tools Appl.* **2018**, *79*, 4315–4331. [CrossRef]
10. Tang, W.; Li, B.; Tan, S.; Barni, M.; Huang, J. CNN-based adversarial embedding for image steganography. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2074–2087. [CrossRef]
11. Kang, Y.; Liu, F.; Yang, C.; Luo, X.; Zhang, T. Color Image Steganalysis Based on Residuals of Channel Differences. *Comput. Mater. Contin.* **2019**, *59*, 315–329. [CrossRef]
12. Shi, L.; Wang, Z.; Qian, Z.; Huang, N.; Puteaux, P.; Zhang, X. Distortion Function for Emoji Image Steganography. *Comput. Mater. Contin.* **2019**, *59*, 943–953. [CrossRef]
13. Jessica, F.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882.
14. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Learning and transferring representations for image steganalysis using convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
15. Jian, Y.; Ni, J.; Yi, Y. Deep learning hierarchical representations for image steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2545–2557.
16. Ye, D.; Jiang, S.; Li, S.; Liu, C. Faster and transferable deep learning steganalysis on GPU. *J. Real-Time Image Process.* **2019**, *16*, 623–633.
17. Zhang, Y.; Zhang, W.; Chen, K.; Liu, J.; Liu, Y.; Yu, N. Adversarial examples against deep neural network based steganalysis. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018.
18. Yang, C.; Wang, J.; Lin, C.; Chen, H.; Wang, W. Locating Steganalysis of LSB Matching Based on Spatial and Wavelet Filter Fusion. *Comput. Mater. Contin.* **2019**, *60*, 633–644. [CrossRef]
19. Schembri, F.; Sapuppo, F.; Bucolo, M. Experimental classification of nonlinear dynamics in microfluidic bubbles' flow. *Nonlinear Dyn.* **2012**, *67*, 2807–2819. [CrossRef]
20. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Bengio, Y. Generative adversarial nets. In Proceedings of the Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
21. Johnson, J.; Alahi, A.; Li, F. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.

22. Snderby, C.K.; Caballero, J.; Theis, L.; Shi, W.; Huszar, F. Amortised map inference for image super-resolution. *arXiv* **2016**, arXiv:1610.04490.
23. Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; Li, H. High-resolution image inpainting using multi-scale neural patch synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6721–6729.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
25. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
26. Goodfellow Ian, J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
27. Jiawei, S.; Vargas, D.V.; Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* **2019**, *23*, 828–841.
28. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, 12–18 October 2008.
29. Patrick, B.; Filler, T.; Pevný, T. Break Our Steganographic System: The Ins and Outs of Organizing BOSS. In Proceedings of the International Workshop on Information Hiding 2011, Prague, Czech Republic, 18–20 May 2011; Springer: Berlin/Heidelberg, Germany, 2011.
30. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017.