A brief introduction to image steganography with GANs

Pascal Huszár

Department for Hardware Oriented Computer Science (HOCOS) Institute for Computing Engineering and Computer Architecture (ITI) University of Stuttgart

Abstract. Encrypting secret messages prevents adversaries from obtaining sensitive information. However, an encrypted message attracts attention, which is not desired when secrets are exchanged. To conceal the secret message itself, image steganography is a technique that attempts to hide the message in images. There are several approaches that use different strategies. The most promising is SteganoGAN by Zhang et al. [1]. Their model is capable of hiding up to 4.4 bits per pixel of arbitrary data inside images while it evades detection by steganalysis tools. On the basis of SteganoGAN, the motives and the concepts of steganographic systems are expressed in this paper.

1 Introduction

Hiding information on a communication channel is desirable if the information or typically the content of a message is solely intended for the sender and receiver. While techniques such as cryptography aims to encrypt a message in such a way that adversaries are no longer able to recover the information, the mere presence of an encrypted message can be suspicious and lures hackers. The goal is to covertly communicate a message. Steganography is the art of hiding the presence of a secret message inside innocuous seeming carriers. While hiding secret information is a well-intentioned application, steganography is also suitable to hide malicious code [2].

Motivation Disguise the mere presence of encrypted messages, decreases the unwanted emerging of attention. Furthermore, a secret encrypted message is only secure as long as the private key is in the hands of the correct owner. Thus, steganography offers additional protection if used in combination with cryptography [3]. There are several domains that would benefit from steganography. In countries where freedom of expression is restricted, secret news and reports can be hidden inside different carriers. The healthcare sector requires a secure and reliable transmission of patient's information and data in which steganographic techniques offer a promising alternative [4] [5]. In the domain of social media and creative content creation, steganography can be used in managing access to content by embedding appropriate copyright data inside the medium [6].

Image Steganography Various types of steganography exist, such as audio steganography, in which hidden information is incorporated into an audio file [7]. Text steganography, a technique that aims to hide text messages inside other text messages [8]. The process of concealing a secret message in a cover image is called image steganography. Whether the secret message is another image or plain text, the goal is that nobody but the recipient or sender will be able to discover the secret message. Furthermore the steganographic image should be indistinguishable from the cover image by the human eye and steganalysis tools. There are different approaches to image steganography that aim to achieve two objectives: To produce real-looking steganographic images along with transmitting a relative high payload. A traditional method to hide data in an image is called LSB-steganography, visualized in Figure 1. The idea is to replace the least

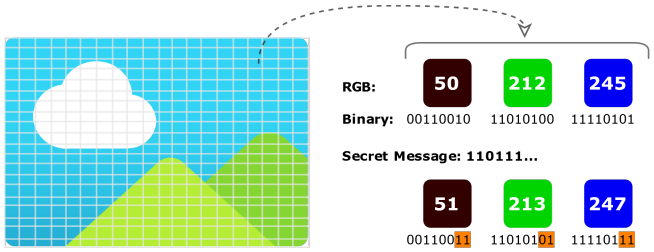

Fig. 1. Least significant bit in digital steganography. To hide the message inside the image the least significant bits are manipulated according to the secret message. The receiver may later recover the hidden message by extracting these bits. The changed color values are practically not visible.

significant bits in an image by the binary secret message. The resulting steganographic image is indistinguishable from the original by the human eye but are easily detectable by automated steganalysis tools. With the enrichment by deep learning methods and computational power, image steganography is undergoing a major boost and new approaches are emerging [1] [9] [10]. While these attempts use neural networks as an end-to-end solution, where the secret message and the cover image are processed into a steganographic image, they encounter certain limitations. Hayes et al. [11] showed robust results with adversarial training on steganographic algorithms but their technique requires the cover image to have a certain defined size. The recent work of Duan et al. [12] utilized a type of DenseNet (Dense Convolutional Network). While their approach achieved a relative high image payload capacity of 23.96 bits per pixel (bpp), only images can be concealed inside other images. These limitations are addressed by the proposed model SteganoGAN by Zhang et al. [1]. Their model is based on the recent success of generative adversarial networks (GANs) in synthesizing realistic images. The novel approach produced realistic steganographic images, embedded arbitrary data inside an image and achieved a relatively high data payload of

4.4 bpp. Therefore, SteganoGAN is a flexible approach to image steganography. To provide an overview of the subsequent sections of this paper, the outline follows. Section 2 describes the approach of image steganography with GANs and, in particular, the approach of SteganoGAN. The countermeasures and detection of steganography systems are introduced in Section 3. The final Section 4 concludes this paper with a conclusion and suggestions for future work.

2 Image Steganography with GANs

The creation of realistic images is a key factor in the obfuscation of a secret message exchange and image steganography in general. In 2014, Goodfellow et al. [13] introduced the generative adversarial network (GAN), a model capable of synthesizing realistic images. The great success of these models motivated further research in utilizing GAN-based systems for the image steganography [9] [1] [11]. The basic concept behind GAN-based steganography systems is the adversarial training of generating realistic steganographic images and the extraction of the hidden message. Adversarial training is a technique in which a model is optimized through the training by extra deceptive input. At best, the model is robust against adversarial inputs or attacks. The encoder or generator (related to GANs) is the first part that attempts to encode as much information as possible while synthesizing realistic images. The contrast is formed by the decoder. It tries to recover the hidden message, embedded from the encoder, as accurate as possible. To improve quality and evade automated detection the critic network or discriminator (related to GANs) evaluates the synthesized images and provides feedback. Recent researches took advantage of GAN-based systems, but had certain limitations [11] [12]. The following section introduces a more flexible approach, called SteganoGAN.

2.1 SteganoGAN

Introduced in early 2019, Zhang et al. [1] reported state-of-the-art performance with their GAN-based approach SteganoGAN for the steganography task of hiding arbitrary data inside images. The following subsections describe the modules of SteganoGAN's architecture, as shown in Figure 2.

Encoder Given some secret message M as a data tensor of shape $D \times W \times H$, where D indicates the number of bits that will be concealed in each pixel and a cover image C with width W and height H. Let T_1 and T_2 be two tensors of same height H and width W but possible different depth $D_{1,2}$. Then, let $f_{+\!\!+} : (T_1, T_2) \longrightarrow \Phi \in \mathbb{R}^{(D_1+D_2)\times W\times H}$ be the concatenation of two tensors along the depth axis. The encoder attempts to hide as much bits as possible in each pixel of C to then generate a realistic steganographic image S. For each cover image C and message M the approach of Zhang et al. [1] applies two preprocessing steps with convolutional blocks:

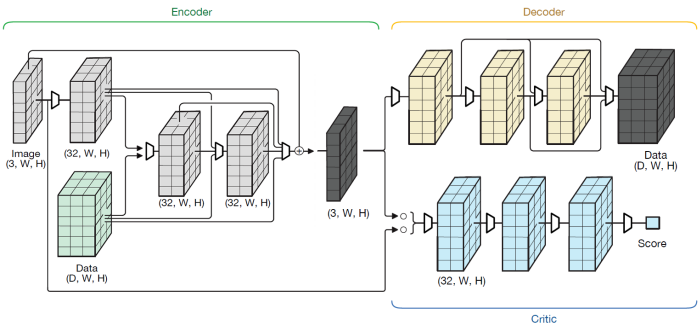

Fig. 2. Model architecture of SteganoGAN with an encoder, decoder and critic network. The dense variant, as one of three different variants of encoders, is visualized.(Zhang et al. 2019 [1])

1. Obtain tensor a given by $a = Conv_{3 \to 32}(C)$
2. Concatenating the message M and tensor a and afterwards obtain tensor b given by $b = Conv_{32+D \to 32}(f_{\#}(a, M))$

After this preprocessing the tensor b is further processed by additional convolutional blocks. The authors presented three different architectures for the encoder. One of these three variants is visualized in Figure 2. The dense variant adds additional connections between the convolutional blocks and subsequent blocks. As stated by the authors, the concatenating of feature maps with these generated by later blocks, mitigates the vanishing gradient problem [1]. The vanishing gradient problem is encountered when the updates for each weight in the neural network getting close to zero and therefore prevent a "learning" of the network. The dense encoder variant can be formally described as:

1. Preprocessing
2. Obtain tensor c given by $c = Conv_{64+D \to 32}(f_{\#}(a, b, M))$
3. Obtain tensor d given by $d = Conv_{96+D \to 3}(f_{\#}(a, b, c, M))$
4. $E_d(C, M) = C + d$
5. $S = E_d(C, M)$

Note that at the last step of the dense encoder the cover image C is added to the tensor d. Result of the encoder is a steganographic image S with the same resolution and depth like the cover image C.

Convolutional Blocks As shown in Figure 2 each module of SteganoGAN consists of several convolutional blocks. In general, convolutions are kernel or filter with learnable parameters that attempts to extract low-dimensional features from the image. For example in face recognition, each layer learns key features from facial images. At best, the last layer learned all key features and is capable of recognizing different faces. In the case of steganography, the convolutional blocks detect spatial features in the image with the hidden message. The blocks

work either on the image or on the output from previous blocks. Through this, the encoder learns the key features to generate realistic steganographic images. In Figure 3, a basic example for the application of a kernel or filter on an image is visualized. The convolutional blocks in SteganoGAN can be formally expressed

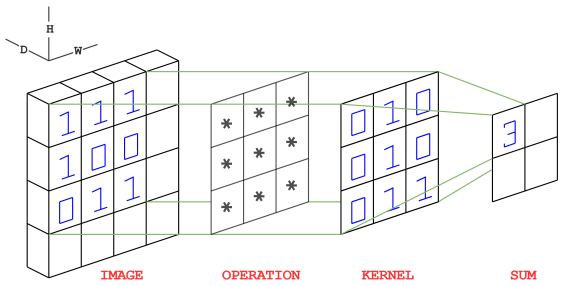

Fig. 3. 2D Convolution. The convolutional filter or kernel moves along two directions (h, w) to calculate low dimensional features from the image data. The kernel can also be of different depth size, meaning the number of depth corresponds to the number of filters applied across the depth of the image. The input multiplied with the kernel forms the activation or feature map.

as: $Conv_{D_1 \to D_2} : T_1 \in \mathbb{R}^{D_1 \times W \times H} \longrightarrow \Phi \in \mathbb{R}^{D_2 \times W \times H}$. The layers of the blocks have a kernel size of 3, i.e. the filter is a matrix with 3×3 size. The filter move by 1 unit, hence a Stride 1. Padding is 'same', i.e. the size of the output does not differ from the input. A leaky ReLU activation function was chosen.

Decoder The network that extracts the hidden message M, concealed in the steganographic image S. Similar to the encoder architecture, the output of a convolutional block is connected to the subsequent blocks. The decoder can be formally expressed as:

1. $a = Conv_{3 \to 32}(S)$
2. $b = Conv_{32 \to 32}(a)$
3. $c = Conv_{64 \to 32}(f_{\#}(a, b))$
4. $D(S) = Conv_{96 \to D}(f_{\#}(a, b, c))$
5. $M' = D_d(S)$

The decoder produces M', therefore attempts to recover the hidden message M as accurate as possible.

Critic Evaluation of the steganographic images generated by the encoder, is beneficial as it provides valuable feedback to the encoder. For this reason, the authors added the critic network. This network consists of 3 convolutional blocks followed by a single convolutional layer to produce a score for the input image. Formally described as:

1. $a = Conv_{32\to32}(Conv_{32\to32}(Conv_{3\to32}(S)))$
2. $C(S) = Mean(Conv_{32\to1}(a))$

$Mean$ is defined as a pooling operation that computes the average of the value in each feature map, produced by the previous convolutional block. Input to the critic network is the cover image C and the steganographic image S. The critic compares both inputs and provides feedback about the realness of S. Typically in GAN models, the discriminator predicts a probability for a certain image of being "real". Recent work, as well as in SteganoGAN, uses a modification of the GAN model, called the Wasserstein GAN (WGAN). It shifts the idea of a discriminator predicting a probability towards a critic network that evaluates the "realness" of a generated image by a score. WGAN model also differs from a "basic" GAN model in the way it uses Wasserstein loss in training the discriminator [14].

2.2 Training

In order to optimize the encoder-decoder network and the critic network, different training strategies are applied. Three questions help to understand these strategies:

- How accurate is the decoder?
- How good is the encoder at synthesizing the original image?
- How realistic is the steganographic image S

Note that all the loss functions are jointly optimized. To optimize decoder's accuracy, the authors used the cross-entropy loss as loss function. Cross-entropy loss measures the difference between the message M' in the steganographic image S and the desired output, message M. To compare the similarity of S and the cover image C and therefore evaluate how good the encoder performs, mean-square error is used. Each pixel in S is compared with the corresponding pixel in C. The higher the difference in total is, the more training is needed to optimize the encoder. Finally, the realness of S is evaluated by the critic network using a Wasserstein loss. This loss function seeks to increase the distance of the measured scores for the cover image C and the steganographic image S. The goal is to encourage the encoder in generating more realistic images, which are then scored higher by the critic network. The overall training objective is to minimize the three losses, outlined by the three questions above.

2.3 Results of SteganoGAN

In Table 1, the results of the dense model variant of SteganoGAN on the COCO [15] dataset is shown. The dataset consists of 330k images with around 1.5M object instances. The data depth D represents the desired bits per pixel in the image. The accuracy is determined by the probability with which the decoder correctly decodes the bits. The structural similarity index or SSIM computes the similarity between the cover image C and the steganographic image S. The function returns a value in the range of $[-1.0, 1.0]$. The value of 1.0 indicates that S and C are identical.

Table 1. Extract from the results of the experiment with the dense model variant of SteganoGAN [1]. The dense variant performed best among all other variants in almost all experimental settings. However, as the number of bits per pixel increases, the accuracy and the recovered payload decreases.

Dataset	D	Accuracy	Bits per pixel	SSIM
COCO	1	**0.99**	0.99	**0.98**
	2	0.99	1.97	0.95
	3	0.98	2.87	0.92
	4	0.95	3.61	0.92
	5	0.92	4.24	0.91
	6	0.87	**4.40**	0.88

3 Steganalysis - Identification of Steganographic Images

At best, the steganography GAN-based model produces realistic looking images with a relative high payload. While these images seem to be indistinguishable by humans, they may be detectable by steganalysis tools. Therefore the target is to evade detection not only by humans but also by steganalysis tools. In order to measure the effectiveness of SteganoGAN, the authors used a popular steganalysis tool called StegExpose [16]. This tool combines different techniques for the detection of steganographic images. For the evaluation, Zhang et al. randomly selected 1000 steganographic images. The authors concluded that the generated steganographic images successfully evaded detection by StegExpose [1]. Deep learning-based approaches for the detection of steganographic images have shown good results in recent works [17] [18]. To evaluate the steganographic images from SteganoGAN by deep learning-based tools they used the model introduced by Ye et al [17]. Zhang et al. found that increasing the number of bits per pixel also increases the detection rate of steganographic images.

4 Conclusions and Outlook

Image steganography systems enable additional protection and privacy from adversaries. There are different approaches for the design of such systems but they focus solely on hiding images in another image or are limited to a small payload. The proposed model SteganoGAN by Zhang et al. is able to hide up to 4.4 bits per pixel of arbitrary data inside an image. This approach provides the flexibility needed, yet produces realistic steganographic images while evading steganalysis tools.

References

1. Zhang, K.A., Cuesta-Infante, A., Xu, L., Veeramachaneni, K.: SteganoGAN: High Capacity Image Steganography with GANs. arXiv:1901.03892 [cs, stat] (2019) arXiv: 1901.03892.

2. Suarez-Tangil, G., Tapiador, J.E., Peris-Lopez, P.: Stegomalware: Playing Hide and Seek with Malicious Components in Smartphone Apps. In Lin, D., Yung, M., Zhou, J., eds.: Information Security and Cryptology. Lecture Notes in Computer Science, Cham, Springer International Publishing (2015) 496–515
3. Taha, M.S., Rahim, M.S.M., Lafta, S.A., Hashim, M.M., Alzuabidi, H.M.: Combination of Steganography and Cryptography: A short Survey. IOP Conference Series: Materials Science and Engineering **518** (2019) 052003 Publisher: IOP Publishing.
4. Douglas, M., Bailey, K., Leeney, M., Curran, K.: An overview of steganography techniques applied to the protection of biometric data. Multimedia Tools and Applications **77**(13) (2018) 17333–17373
5. Sahu, N., Peng, D., Sharif, H.: An innovative approach to integrate unequal protection-based steganography and progressive transmission of physiological data. SN Applied Sciences **2**(2) (2020) 237
6. Shih, F.Y.: Digital Watermarking and Steganography: Fundamentals and Techniques, Second Edition. CRC Press (2017) Google-Books-ID: j4ujDgAAQBAJ.
7. Gopalan, K.: Audio steganography using bit modification. In: 2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698). Volume 1. (2003) I–629
8. Hamdan, A.M., Hamarsheh, A.: AH4S: an algorithm of text in text steganography using the structure of omega network. Security and Communication Networks **9**(18) (2016) 6004–6016 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1752.
9. Qin, J., Wang, J., Tan, Y., Huang, H., Xiang, X., He, Z.: Coverless Image Steganography Based on Generative Adversarial Network. Mathematics **8**(9) (2020) 1394 Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.
10. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: HiDDeN: Hiding Data With Deep Networks. arXiv:1807.09937 [cs] (2018) arXiv: 1807.09937.
11. Hayes, J., Danezis, G.: Generating Steganographic Images via Adversarial Training. arXiv:1703.00371 [cs, stat] (2017) arXiv: 1703.00371.
12. Duan, X., Nao, L., Mengxiao, G., Yue, D., Xie, Z., Ma, Y., Qin, C.: High-Capacity Image Steganography Based on Improved FC-DenseNet. IEEE Access **8** (2020) 170174–170182 Conference Name: IEEE Access.
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: (Generative Adversarial Nets) 9
14. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv:1701.07875 [cs, stat] (2017) arXiv: 1701.07875.
15. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., DollÃ¡r, P.: Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs] (2015) arXiv: 1405.0312.
16. Boehm, B.: StegExpose - A Tool for Detecting LSB Steganography. arXiv:1410.6656 [cs] (2014) arXiv: 1410.6656.
17. Ye, J., Ni, J., Yi, Y.: Deep Learning Hierarchical Representations for Image Steganalysis. IEEE Transactions on Information Forensics and Security **12**(11) (2017) 2545–2557 Conference Name: IEEE Transactions on Information Forensics and Security.
18. Khan, N., Haan, R., Boktor, G., McComas, M., Daneshi, R.: Steganography GAN: Cracking Steganography with Cycle Generative Adversarial Networks. arXiv:2006.04008 [cs] (2020) arXiv: 2006.04008.