Towards Generative Image Steganography using Seq2Seq and GAN

Anditya ARIFIANTO[1,2,a)] Takahiro KOMAMIZU[1,b)]
Yasutomo KAWANISHI[3,c)] Ichiro IDE[1,d)]

Abstract

With the increasing attention to research on coverless image steganography, we have re-explored the use of Generative Adversarial Network (GAN) in generating images to hide messages that are effective yet simpler to use. Compared to traditional image steganography which embed the secret message into an image, this Generative Image Steganography technique transforms the message directly into a generated image called stego-image.

Our technique uses a sequence encoder to read a text message and a generator encoder to convert it into an image. To retrieve hidden messages, we use a sequence decoder and a generator decoder. All models are trained in an end-to-end fashion. Our technique is able to produce generated stego-images with an Fréchet Inception Distance (FID) of 0.13, and is able to transform it back into text messages with a BLEU metric of 0.59 and a Word Error Rate of 0.26.

1. Introduction

With the increasing quality and speed of Internet connectivity, people are now able to transmit ever-larger amounts of data. Which is why in recent years, people send images very often when communicating other than just sending text messages. Thus, sending pictures to each other has become a natural task in the eyes of the society, whether the image actually means something or just a random *meme*.

On the other hand, the rapid development of surveillance technology and the growing threat to information security have increased the attention to the protection of personal communications. One of the popular techniques for securing messages from unwanted readers is to hide text messages into images which is called image steganography [1]. The technique is done by taking an image to cover the message, called a cover image, then insert the message bits into the pixels, resulting in a stego-image which contains the secret message. The goal is to trick anyone intercepting the message into thinking that the image sent is just a plain image that does not contain anything suspicious. However, the traditional image steganography method generally results in some image distortion that can cause the presence of hidden information to be successfully detected by a Steganalyzer [2].

Steganalysis is a technique that aims to detect the presence of hidden information in the stego-image. Various methods have been proposed for steganalysis detection which combine handcrafted feature extraction with machine learning algorithm [2]. Although some conventional steganographic approaches can deceive one or several steganalyzers, it is very difficult to resist the detection of a well-designed one [3,4]. The main reason is that the conventional steganographic operation, which mainly embeds the secret message by slightly modifying some insensitive image features, will inevitably cause some distortion in the stego-image, especially under high message payload. Such distortion will cause the presence of hidden information to be detected.

Meanwhile, recent developments in the field of image generation have prompted the development of another paradigm called Generative Image Steganography [1, 5]. Compared to most traditional image steganography which embed the secret message into a cover image, this technique transforms the message directly into a generated image. This makes the technique more resistant to steganalysis detection because the resulting image is generated from scratch, so there is no trace of image manipulation as occurred in traditional image steganography [6–9].

In this presentation, we further explored the Generative Image Steganography based on sequence processing and Generative Adversarial Network (GAN) [10]. We are pursuing the intention to improve the stegosampling algorithm to produce a simple but effective solution for Generative Image Steganography. For this, we propose a method combining Seq2seq processing with GAN to convert input text into generated images to hide the message contents. We will also introduce our challenge to implement the concept of private key into the proposed method.

2. Related Work

2.1 Image Steganography

To date, many conventional image steganography approaches have been proposed. They generally use an existing

[1] Nagoya University
[2] Telkom University
[3] RIKEN
[a)] anditya@telkomuniversity.ac.id
[b)] taka-coma@acm.org
[c)] yasutomo.kawanishi@riken.jp
[d)] ide@i.nagoya-u.ac.jp

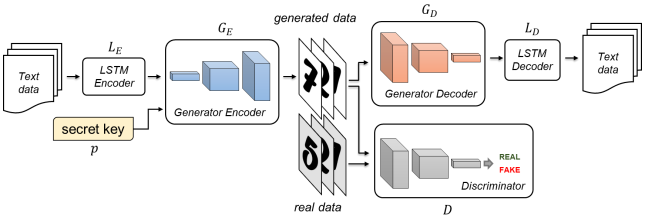

Fig. 1 Generative Steganography with Seq2Seq and GAN training

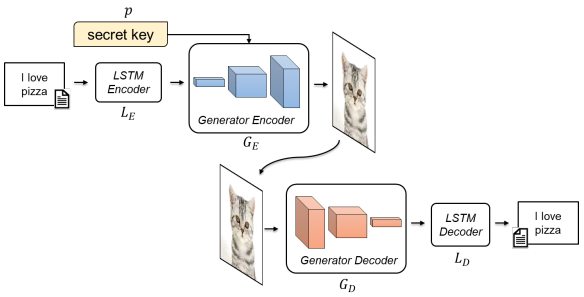

Fig. 2 Proposed framework

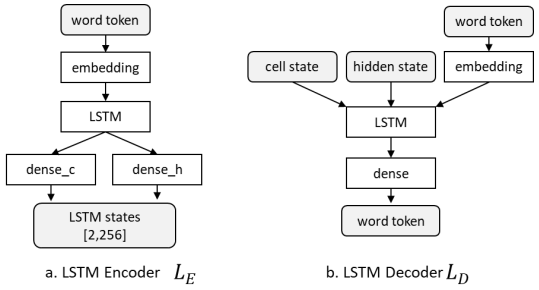

Fig. 3 Seq2Seq Architecture

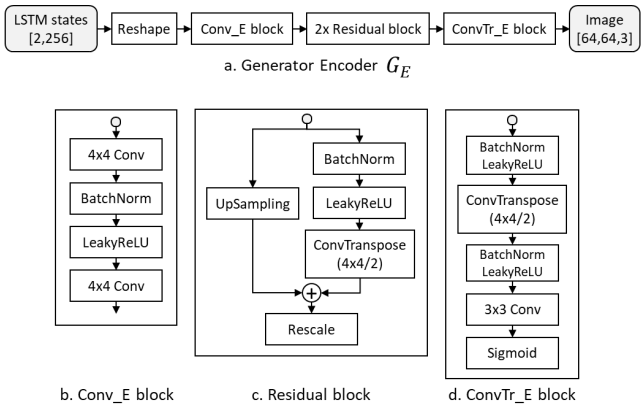

Fig. 4 Generator Encoder Architecture

image as the cover and then imperceptibly embed confidential information into the cover image by slightly modifying the image pixels. The most basic technique is to modify the Least Significant Bit (LSB) of each pixel of the cover image with the same probability to insert a secret message [11].

However, this modification method inevitably leaves some traces of modification and distortion in the original image. Therefore, we face such a problem that this steganographic method cannot resist the detection of existing steganalytic tools. Since then, many techniques have been proposed to improve the image modification function to apply steganography with less image distortion [12, 13].

2.2 Generative Image Steganography

Generative Image steganography that transforms secret information to a new generated image that looks very realistic has shown a promising result as a technique to resist steganalysis detection [6]. The idea of using generative models such as GANs in Steganography is to produce disparity of results. This is done to overcome the weakness of Deep Learning which is still deterministic. Thus, if applied to the encryption technique, one can map the decrypted output message to the original input. By introducing the idea of a generative model, the results of an encrypted message can vary which will make it more difficult to map and decrypt.

Some approaches usually convert the secret message into a simple image by inserting it into a latent vector, and turning it into a realistic image with a GAN generator [7, 14]. However, this approach is applicable only to a short secret message. Therefore, several recent developments have tried to use more complicated techniques in the embedding process to increase the capacity of steganography [8, 9].

Most of the techniques using GAN only allow one-way mapping from input information to image content. Message extraction are usually performed by training other models as decoders separately. Another problem is that most existing techniques do not implement any private or secret key mechanism. It means that anyone with access to the decoder model will be able to read the hidden message.

3. Proposed Framework

3.1 Generative Steganography Framework

We propose a method to secure information from input text messages by disguising it as an image using Seq2Seq [15] to encode the input text and GAN to generate the stego-image. Our proposed framework combines the simplicity and processing capabilities of Sequence to Sequence and Generative Models which is effective yet simpler compared to other approaches. The framework is shown Fig. 1. First, the input text is encoded using the Long Short-Term Memory (LSTM) [16] Encoder model L_E which converts the input text into a sentence embedding. The sentence embedding is then fed into the Generator Encoder G_E along with the private or secret key p resulting in new images being generated.

The second generator or the Generator Decoder G_D receive the generated images and is trained to convert it back into the sentence embedding. The generated images are also fed into the Discriminator D to check whether it can still be recognized as unrealistic images. The retrieved sentence embedding then can be read back as the original text using the LSTM Decoder L_D. All of the networks are trained in an end-to-end fashion.

While Generator Encoder is a generally used function, Generator Decoder can be specially trained based on a secret key for a single intended recipient. Therefore, after the

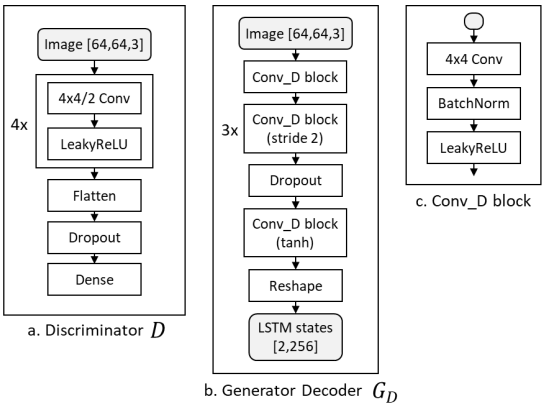

Fig. 5 Generator Decoder Architecture

model is trained, the sender can use the Encoder Generator to generate an image for a specific recipient according to the receiver's secret key. The image then can be sent and can only be read back by the correct Generator Decoder owned by the recipient. The proposed framework is shown in Fig. 2.

3.2 Model Architecture

For the LSTM Encoder, we use an LSTM layer with embedding layer. The cell state and the hidden state of the LSTM is continued to a dense layer for linear transformation into a 256-dimensional vector each as depicted in Fig. 3a.

The Generator Encoder receives both states and reshapes it into a $(8, 8, 8)$ tensor, before it is fed into a Convolution block. The output then proceeds to two Residual blocks and a final Transpose Convolution block. The complete architecture can be seen in Fig. 4.

The Discriminator model consists of four layers of 4×4 stride 2 Convolution, and a Dense layer at the end. The architecture can be seen in Fig. 5a. Similarly, the Generator Decoder also consists of four layers of Convolution block. The difference with the Discriminator is the addition of Batch Normalization layer into each block. The output is then continued to a final convolution block with hyperbolic tangent (tanh) activation and then reshaped to a size $(2, 256)$ array as shown in Fig. 5b.

The LSTM Decoder receives the output from the Generator Decoder as two vector states and feed them into an LSTM layer with a dense layer for word prediction as depicted in Fig. 3b.

4. Evaluation

4.1 Generative Steganography Experiments

We conduct our experiments using the CelebA dataset [17] to produce "realistic-looking" images. For the sake of training efficiency, we rescale the sizes of images to 64×64 pixels and train the model on the rescaled images. For the payload, we use English texts with a length of up to 30 words per sentence taken from the Tatoeba dataset [18].

We trained our model end-to-end with 100,000 sentences and 162,770 images for training and 20,000 sentences for testing. For the evaluation criteria, the BLEU metric [19] and the Word Error Rate are used to measure the retrieval

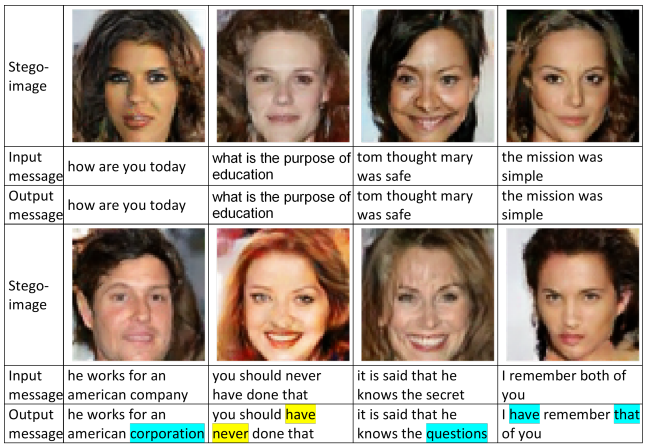

Fig. 6 Example of the generative steganography result.

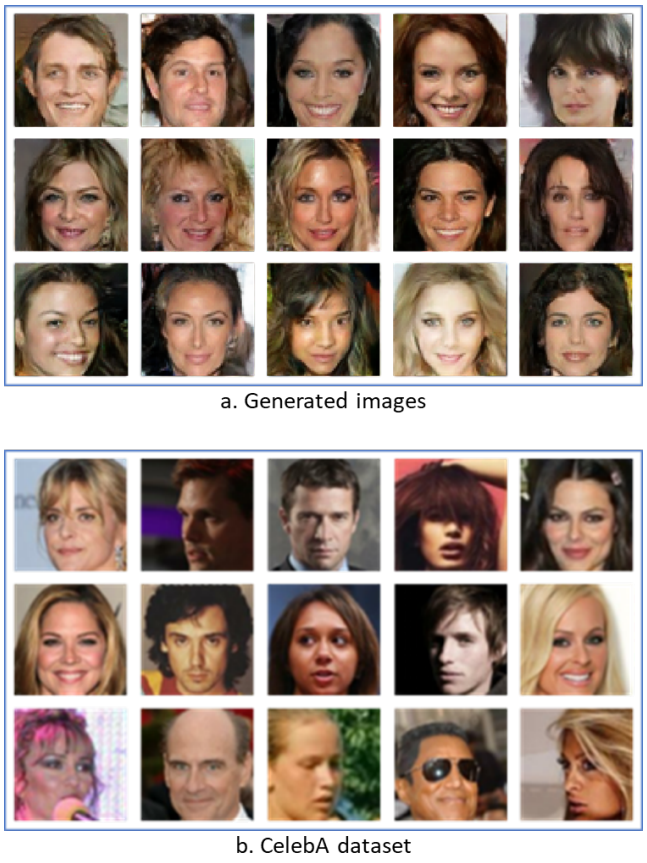

Fig. 7 Comparison of (a) the generated images and (b) CelebA dataset

performance, and the Fréchet Inception Distance (FID) to measure the image quality.

4.2 Steganography Results

Our initial observations yielded moderate but encouraging results. The proposed model achieve 0.59 in BLEU metric and 0.26 Word Error Rate when retrieving text messages. Examples of the input message, the resulting stego-image, and the retrieved message can be seen in Fig. 6.

We can see that the proposed method is able to produce a stego-image with a fairly good performance, but there are still some word prediction errors that must be corrected.

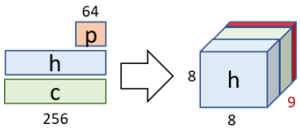

Fig. 8 Secret Key mechanism

Fig. 9 Images generated from the same sentence with different secret keys

Improvements to increase the length of sentences that can be stored also still need to be considered in order to accommodate longer messages.

For image quality, the poposed generator is capable of producing images with an FID score of 0.134. As we can see in Fig. 7, the resulting stego-image, in comparison with the original CelebA dataset is quite good.

5. Challenge on Secret Key Addition

Our next goal is to add a secret key mechanism to the framework so that the same message will be generated into different stego-images depending on the secret key entered. We are considering to add a unique 64-dimensional vector to the Generator Encoder along with the LSTM Encoder states. Together with the LSTM states, the secret key is reshaped into a $(8, 8, 9)$ tensor as shown in Fig. 8.

The implementation stage is to train the entire model as a base model with random secret keys. After trained, for each user, a Generator and LSTM Decoder will be fine-tuned with the designed secret key.

However, as shown in Fig. 9 this mechanism has shown not to work well because the images generated from the same sentence based on different secret keys did not produce significantly different stego-images. This could be due to the length of the secret key being too small when entered into the Generator Encoder. Thus, the secret key only acts as noise instead of a mechanism to stylize the output image.

Therefore, further improvements on the model architecture and training scheme still need to be made to achieve this goal. One solution that can be tried is to use an architecture such as StyleGAN2 [20] to stylize the output image.

6. Conclusion

We propose a technique to perform Generative Image Steganography using Seq2Seq and GAN which are trained in an end-to-end fashion. The proposed model was able to generate stego-images with an FID of 0.13, and is able to transform it back into text messages with a BLEU metric of 0.59 and a Word Error Rate of 0.26.

We also discussed the challenge to add a secret key to the framework to further secure the message. However further effort still needs to be made to realize this function.

References

[1] R. Mishra and P. Bhanodiya, "A review on steganography and cryptography," in *Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications (ICACEA)*, pp. 119–122, 2015.

[2] A. Selvaraj, A. Ezhilarasan, S. L. J. Wellington, and A. R. Sam, "Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning based techniques," *IET Image Processing*, vol. 15, no. 2, pp. 504–522, 2021.

[3] J. Qin, Y. Luo, X. Xiang, Y. Tan, and H. Huang, "Coverless image steganography: A survey," *IEEE Access*, vol. 7, pp. 171372–171394, 2019.

[4] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "Cnn-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, 2019.

[5] J. Liu, Y. Ke, Z. Zhang, Y. Lei, J. Li, M. Zhang, and X. Yang, "Recent advances of image steganography with generative adversarial networks," *IEEE Access*, vol. 8, pp. 60575–60597, 2020.

[6] K.-C. Wu and C.-M. Wang, "Steganography using reversible texture synthesis," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 130–139, 2015.

[7] A. Arifianto, M. A. Maulana, M. R. S. Mahadi, T. Jamaluddin, R. Subhi, A. D. Rendragraha, and M. F. Satya, "EDGAN: Disguising text as image using generative adversarial network," in *Proceedings of the 8th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–6, 2020.

[8] S. Zhang, Z. Yang, H. Tu, J. Yang, and Y. Huang, "Pixelstega: Generative image steganography based on autoregressive models," *Computing Research Repository, arXiv preprint*, arXiv:2112.10945, 2021.

[9] Z. Zhou, Y. Su, Q. M. J. Wu, Z. Fu, and Y. Shi, "Secret-to-image reversible transformation for generative steganography," *Computing Research Repository, arXiv preprint*, arXiv:2203.06598, 2022.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, 2014.

[11] C.-K. Chan and L. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[12] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proceedings of the 12th International Conference on Information Hiding (IH)*, pp. 161–177, 2010.

[13] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proceeding of The 2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4206–4210, 2014.

[14] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38303–38314, 2018.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, vol. 2, pp. 3104–3112, 2014.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," *Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

[18] J. Tiedemann, "The Tatoeba translation challenge – Realistic data sets for low resource and multilingual MT," *Computing Research Repository, arXiv preprint*, arXiv:2010.06354, 2020.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 311–318, 2002.

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8110–8119, 2020.