IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 36, No. 4, August, 2022, pp. 561-567
Journal homepage: http://iieta.org/journals/ria

GAN-Based Encoding Model for Reversible Image Steganography

Afolashade Kuyoro, Uchenna J. Nzenwata*, Oludele Awodele, Sunday Idowu

Computer Science Department, Babcock University, Ilishan-Remo, 121003, Ogun State, Nigeria

Corresponding Author Email: nzenwatau@babcock.edu.ng

https://doi.org/10.18280/ria.360407

Received: 26 April 2022
Accepted: 6 July 2022

Keywords:
reversible image steganography, similarity index, GAN, payload capacity, cover image, secret image

ABSTRACT

In carrying out reversible image steganography, the Generative Adversarial Networks (GANs-based) models have proven to be the most suitable deep learning models for image steganography. Image steganography is a steganography system that hides secret data in an image cover medium without arousing suspicion, and it is defined by the ability to reconstruct the cover medium with no visible distortion after the steganography system has been decoded by extracting the hidden data. In this study, we try achieve the encoding phase in image steganography, where two GAN-base models (CycleGAN and DCGAN) were proposed. Empirical analysis was done to determine a better model for the encoding of image steganography. The Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index Metric (SSIM), and bit per pixel (bpp) were used as the metrics for the analysis. The outcome of DCGAN yielded (SSIM=0.48; PSNR=19.86; bpp=24.79) and the outcome of using CycleGAN yielded (SSIM=0.97; PSNR=41.45; bpp=24.97). These values concluded that the CycleGAN was preferable over the DCGAN. Hence, the CycleGAN was adopted as the encoding model.

1. INTRODUCTION

Data concealing, also known as information hiding, is a common information security approach used in covert communication, digital copyright protection, and other situations [1]. Following the study [2, 3], if we choose to categorise data concealing based on how retrievable it is, we can split it into two types: irreversible data hiding (IDH) and reversible data hiding (RDH). Traditional data concealing methods fall under the first category, whereas the second category is a unique approach used mostly in the medical, legal, and military domains [4, 5].

In any case, every type of image steganography will always use a cover image to conceal secret information [6]. Also, the types of secret information that may be concealed vary: text, images, audio files, and video files. In a survey of image format steganography done in the study [7], it is identified that the payload capacity required for effective steganography has an impact on certain types of hidden information, and text and image format secret information are the most prevalent types of secret information hidden using image steganography [8]. Traditional approaches, such as spatial and transform domain techniques, have been used for a long time to hide secret information [9], mainly text in images, but when images are hidden in another image known as the cover image, these methods might appear sophisticated, thereby making the stego image distorted, and this raises a red flag [9].

In the survey and analysis of various steganographic techniques [10], Least Significant Bit (LSB) was identified as the simplest of all spatial domain techniques used in steganography. Also, the study identified Discrete Fourier Transformation technique (DFT), Discrete Cosine Transformation technique (DCT) and, Discrete Wavelet Transformation (DWT) as the type of transform domain techniques. These techniques are referred to as traditional steganography techniques [11-14], and are characterized by poor payload capacity. That is, the amount of secret information (in bit size) that can be hidden is relatively small.

In order to enhance the payload capacity and reduce the complexity in the traditional methods, deep learning methods was first used in the steganalysis (the detection of hidden information from the steganography system), by suing Convolutional Neural Network (CNN) [15, 16]. Also, Baluja [17], implemented the first deep learning technique to prototype the LSB technique. The study a deep learning approach that compresses and distributes the secret image's representation across all of the available bits in the cover image. This paved way for exploring steganography using deep learning tools.

In regardless of the techniques used, image steganography has always been faced with challenges relating to payload capacity, security and robustness. Some approaches like the steganoGAN [18], encoder-decoder: DCGAN [19], HidingGAN [20] and many more have been adopted to solve these inadequacies, but ended in trade-off between the payload capacity and the security of image steganography systems. Zhang et al. [21] suggested that these in adequacies could be solved when an appropriate cover is selected and used for a targeted secret image. He went further to implement cover selection model that would aid the selection of appropriate cover images for targeted secret images.

This article is an excerpt from a full thesis in which a hybrid deep learning model was presented to improve the performance of existing reversible image steganography methods. Therefore, it focuses on the encoding phase of the reversible image steganography. We proposed two deep learning models (DCGAN and CycleGAN) from the characterization of the existing models built using Generative

Adversarial Network (GAN-base) models.

2. LITERATURE REVIEW

A novel image Steganography Without Embedding (SWE) was proposed using Deep Convolutional Generative Adversarial Networks (DCGAN) [22]. The study used the generator from the Generative Adversarial Network (GAN) to generate the cover image; and the secret message was mapped into the noise vector of the generated cover image. The experimental result shows advantages of highly accurate information extraction and a strong ability of the steganography system to resist attacks because of the CNN. It was discovered that the study only focused on securing the steganography system, but the payload capacity was not encouraging, and the steganography was irreversible. In overcoming the gaps in SWE method, Tang et al. [23] presented a steganographic model with Adversarial Embedding (ADV-EMB). ADV-EMB model achieved the goal of hiding a stego message while at the same time fooling a Convolutional Neural Network (CNN) based steganalyzer. It achieved better security performance against CNN's steganalyzer by increasing its missed detection rate, but still leaves stego images with minute traces of distortions and was not reversible. Dan et al. [24] considered the idea of reversible steganography as a good tool that will enhance a steganography system. In their study, an image steganography system was implemented with the aim of having a good reversibility steganography. By using a U-Net structured Convolutional Neural Network [24], realized that both the cover and the secret images can be concatenated into a 6-channels tensor as an input to the hiding network thereby making reversibility possible. The result of the study shows that GAN and CNN are good deep learning tools for a reversible steganography, but the study did not take into consideration the payload capacity as GANs are not known to improve the embedding capacity of steganography system. This was also confirmed by Zhang et al. [25] that image steganography with GANs is not capable of developing image steganography systems with good payload capacity, except an appropriate cover image can be intentionally selected before the encoding process.

SteganoGAN was a tool developed by Duan et al. [26], to improve on the capacity using GANs which optimizes the perception quality of the stego-images, but the security of the system tested poor in the face of a steganalyzer. To improve the security in SteganoGAN, Ray et al. [27] made use of SegNet structure (a fully convolutional neural networks for image segmentation) to achieve a high-capacity reversible image steganography. The outcome of their study yielded good payload because a cover selection process was done. Also, a Convolutional Neural Network (CNN) with a Deep Supervision-based Edge Detector (DSED) was used by Kadhim et al. [28], and the experimental outcome yielded a good payload capacity due to the edge detector model used for cover selection. This idea was borrowed from Byrnes et al. [29], where it was suggested that classification methods could be used in making quality cover selection for image steganography. But the study yielded poor security. Liu et al. [30] gave a suggestive idea that Cover-Dependent Data Hiding with a cover preparation network (DDH with P), Cover-Dependent Data Hiding without a cover preparation network (DDH without P), and Universal Deep Hiding (UDH) are three meta-architectures that could be useful in solving steganography problems of capacity and security by choosing appropriate Hybrid techniques. The choice of their model was informed based on the results of the studies from Chen et al. [31, 32]. Chen et al. [31] adopted Cycle-consistence Generative Adversarial Network (CycleGAN) as the DDH with P for the preparation and selection of cover image, and deep convolutional generative adversarial network (DCGAN) as the UDH for the encoding phase. The outcome of this study actually gave a positive pull towards achieving the basic features of a good steganography, but was not able to achieve good payload capacity. Petzka et al. [32] was able to achieve good payload capacity by using a non-linear Support Vector Machine (SVM) to select appropriate cover image and three convolutional layers with different kernels and Gaussian noise for the security of the system. The outcome was so much encouraging but the study was carried out with grey colored images.

3. METHODOLOGY

The encoding phase is the concealing network that creates the stego-images or carrier images. To achieve this phase, this study proposes the Cycle-Consistent Generative Adversarial Network (CycleGAN). The CycleGAN was suggested since it can train flawlessly without requiring samples of the translated image. Additionally, it includes a wide range of training options, ensuring a model that fits correctly. Also, the CycleGAN structure can prevent model overfitting and help to reduce consistent loss in the image bits. The encoding model receives cover images and the corresponding secret images as input, as shown in Figure 1.

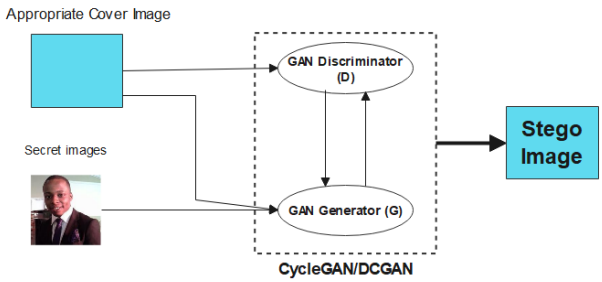

Figure 1. CycleGAN/DCGAN encoder model

The cover image selection model (DNN/SVM model) is passed alongside with the secret information into the CycleGAN/DCGAN's model generator G for the encoding process. After a successful encoding, the stego image is discriminated with the cover image using the encoding model's discriminator (D). This continues until the discriminator fails to discriminate between the original cover image and the stego image. The stego image is then sent as output from the encoding phase. To justify the choice of CycleGAN model for the encoding model, this study also considered the training of Deep Convolutional Generative Adversarial Network (DCGAN). The model with a better payload and security was adopted. The general GAN's loss equation described in Eq. (1) and (2) [33, 34] was used to optimize the losses in the encoder's network.

$$Loss = Min_{(G)}Max_{(D)}[\log(D_{(x)}) + \log(1 - D(G_{(z)}))] \quad (1)$$

where, Eq. (1) was used by considering a single data point. To consider the entire data set, Eq. (1) is transformed to Eq. (2):

$$Min_{(G)}Max_{(D)}V(D,G) = Min_{(G)}Max_{(D)}\left(E_{x\sim P_{data(x)}}[log\,(D_{(x)})\right] + E_{z\sim P_{data(z)}}[\log(1-D(G_{(z)}))]) \quad (2)$$

The stego image was analysed to obtain the payload capacity and the security of the steganography system.

3.1 Evaluation metrics

Because the Generative Adversarial Networks (GANs) was used in this study as the fundamental encoding blocks, the models were evaluated using image quality metrics (also known as the denoising techniques), as GANs do not have objective evaluation functions as supposedly stated in the paper [34]. The proposed image quality metrics are as follows:

3.1.1 Peak signal-to-noise ratio (PSNR)

This metric is for improving the perceptual quality in stego-images (C'). This metric was used in this study to compare the quality of the stego-image (C') to its matching cover image (C) by measuring the peak signal-to-noise ratio of two images, (C') and (C). The greater the PSNR in decibel (dB), the higher the visual quality. It is calculated using the following Equation:

$$PSNR_{(C',C)} = 10log_{10}\left(\frac{Max^2C}{MSE}\right) \quad (3)$$

Taking the square root of the Eq. (3),

$$PSNR_{(C',C)} = 20log_{10}\left(\frac{MaxC}{\sqrt{MSE}}\right) \quad (4)$$

$$= 20log_{10}(Max^2C) - 20log_{10}(MSE_{C',C}) \quad (5)$$

$$MSE_{(C',C)} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(C(i,j)-C'(i,j))^2 \quad (6)$$

where, *MSE* is the Mean Squared Error, and it is given by, *C*=the matrix data of our cover image; *C'*=the matrix data of our stego-image; *m*=the numbers of rows of pixels of the images and i represents the index of that row; *n*=the numbers of columns of pixels of the images and j represents the index of that column; *maxc*=the maximum signal value that exists in the cover image.

The PSNR values of the HDLM was used to ascertain the level of the system's security. These values were compared with the PSNR values of the existing models.

3.1.2 Structural similarity index (SSIM)

SSIM is for regularizing parameter selection in image restoration with inverse gradient. This metric is proposed in this study, in case there is an undefined value of PSNR, where the MSE = 0, and this makes human perception not important. As such, SSIM is necessary, since it aids the Human Vision System (HVS). The SSI calculates the degree of similarity between two images. It is regarded as one of the most often used quality metrics, and it is connected to the single-scale measurement, which performs best when used at an appropriate scale. The best denoising approach is indicated by the highest SSIM.

SSIM was used to compare corresponding pixels and their neighborhoods in the cover and the stego-images, denoted by C and C', using three quantities: luminance (I), contrast (C), and structure (S). The equation is given thus:

$$I(C,C') = \frac{2\mu_C\mu_{C'}+k_1}{\mu_C^2\mu_{C'}^2+k_1} \quad (7)$$

$$C(C,C') = \frac{2\sigma_C\sigma_{C'}+k_2}{\sigma_C^2\sigma_{C'}^2+k_2} \quad (8)$$

$$S(C,C') = \frac{\sigma_{CC'}+k_3}{\sigma_C\sigma_{C'}+k_3} \quad (9)$$

where, the variables $\mu_C, \mu_{C'}$, σ_C and $\sigma_{C'}$ are the mean and standard deviations of the pixel intensity in a small image patch centered on *C* or *C'*. The variable σ_{CC}' represents the sample correlation coefficient between matching pixels in patches centered on C and C'. k_1, k_2, and k_3 are minor values that were included for numerical stability.

To derive the SSIM equation, Eqns. (7-9) were combined to produce Eq. (10):

$$SSIM(c,c') = [l(c,c')^{\alpha}].[C(c,c')^{\beta}].[S(c,c')^{\gamma}] \quad (10)$$

where, α, β and γ are the positive constants that must be greater than zero (α, β, γ>0).

3.1.3 Payload capacity

In other to calculate the size of concealable secret message and maximize the embedding capacity of the steganography system, this study employed the bit per pixel (BPP) method as shown in Eq. (11):

$$BPP = \frac{Number\ of\ secret\ bits\ embeded}{Total\ pixel\ on\ the\ cover\ image} \quad (11)$$

Traditional state-of-the-art steganography techniques has a maximum of 4.4bpp embedding capacity [25]. This study is expected to have a higher embedding capacity based on the proposed cover selection model.

3.2 Dataset

This study used image dataset from the Microsoft Common Objects in Context (COCO) dataset. Microsoft COCO dataset is the benchmark for assessing the performance of deep learning computer vision models. It consists of 328,000 images, 883,331 object annotations, 80 classes, and image ratio of 512×512. This study used the 2017 version of Microsoft Common Objects in Context (COCO) unlabeled dataset of over 250, 000 images.

4. ENCODING MODEL

The stego-image was created using the encoding model phase. In encoding the secret image into an appropriate cover image, this study used two deep learning models: Cycle-Consistent Generative Adversarial Network (CycleGAN) and Deep Convolutional Generative Adversarial Network (DCGAN). This was done in order to assure a more accurate encoding model. The reason for using these two models was based on their frequent use as the common Generative Adversarial Networks (GANs) best used for image translation and generation. CycleGAN and DCGAN were adapted in this

study to function as image steganography encoders by stacking the cover image with the secret message so that the cover image covers the secret message after an acceptable number of epoch training. When the cover image entirely overlaps the hidden message, the maximum training epoch is attained, and a stego-image comparable to the original cover image is created.

4.1 Training the CycleGAN

This study implemented the cycleGAN encoder model by importing the generator and discriminator used in the pix2pix pretrained model. This was achieved by installing the tensorflow_examples package. This is because the cycleGAN encoder model architecture is similar to pix2pix model. Also, there is no paired data to train on in CycleGAN, hence there is no assurance that the input translated secret image and targeted stego-image are meaningful before training as shown in Figure 2.

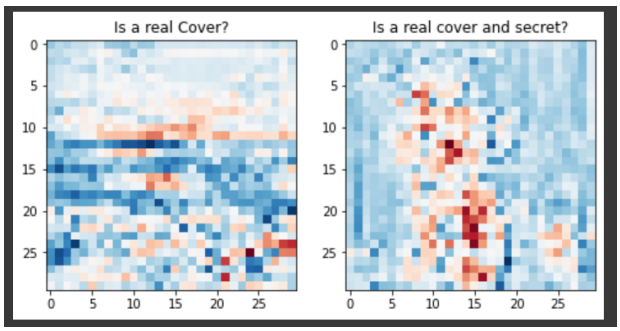

Figure 2. The output from the Encoder's untrained model showing no meaningful outcome

Therefore, there was the need to perform some image augmentation techniques by applying random image jittering and image mirroring on the training dataset. This was necessary to avoid overfitting or underfitting of the model. Figure 3 shows a sample of the cover and secret images after applying random jittering and mirroring. Random image jittering was done by upscaling the images to 286×286×3 from 64×64×3, then resizing to 256×256×3. While image mirroring was done by randomly flipping the images horizontally from left to right. This was necessary so to ensure quality stego-image and also to maintain tolerable training time. After that, the image training dataset was normalized using [-1, 1]. This was done to ensure that the model trains faster and ensure similar pixel distribution.

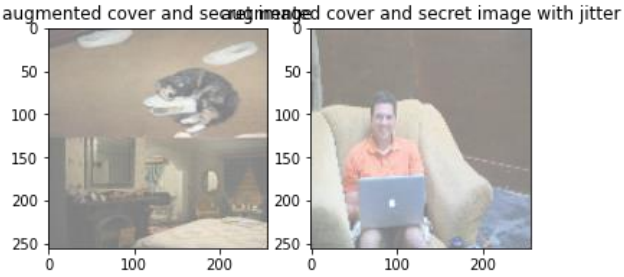

Figure 3. Augmented cover and secret images with jitter

There are two generators and two discriminators in cycleGAN. In training our cycleGAN encoder model, one generator and one discriminator were used. This is because at this point, we do not consider using cycleGAN as a decoder. It is therefore necessary to use one generator to generate the stego-image and one discriminator to determine a fully trained encoder model.

A modified resnet-based generator was employed, and the instance normalisation approach was used for the feature normalisation.

The cycle consistency loss given in Eq. (12), was used to calculate the generator and the discriminator losses of the cycleGAN model.

$$cycle\ consistency\ loss = \lambda \mathrm{Lcyc}(\mathrm{G}, \mathrm{F}) \tag{12}$$

where, λ=lamda constant which was set to be=10; G=generator loss; F=discriminator loss.

Adam optimizer as the optimization function for reducing the model loss and improve the training rate of the model. The training batch size was set to 1 with a learning rate of 0.0002. The activation function used was LeakyRelu because it is faster to compute and makes provision for a small negative slope value, unlike the Relu function. The model was set to train for 210 epochs. Table 1 shows the summary of the cycleGAN training hyper-parameters.

Table 1. CycleGAN training hyper-parameters

SN	Hyper-Parameters	Values
1.	batch_ size	160
2	Activation function	LeakyRelu
3.	Learning_rate	0.0002
4.	epoch	185
5.	loss_function	Cycle consistency loss
6	Lamda (λ)	10
7.	Model optimizer	Adam(2e-4, beta_1=0.5)

4.2 Training the DCGAN

DCGAN was used in like manner as the cycleGAN. These two models are used for image-to-image translation, and as a result was adopted in this study to serve as an alternative encoder model. Using the same dataset, the DCGAN encoder model takes similar parameters as that of the cycleGAN, but with differing values. The DCGAN uses a U-Net-based generator architecture unlike the cycleGAN which uses the renet-based generator. Also, in implementing the DCGAN encoder model, three (3) convolutional-2D layers were used, and the model features were normalized using the batch normalization method. Relu and Tanh were used as the activation function at different layers of the DCGAN encoder model. Table 2 summarizes the hyper-parameters settings of the DCGAN encoder model.

Table 2. DCGAN training hyper-parameters

SN	Hyper-Parameters	Values
1.	batch_ size	255
2	Activation function	LeakyRelu/Relu/Tanh
3.	Learning_rate	0.002
4.	Epoch	210
5	Lamda (λ)	100
5.	loss_function	L1-loss
6.	Model optimizer	Adam (2e-4, beta_1=0.5)

4.3 The encoder model algorithm

In order to carry out the encoding process using the proposed encoding model, a pair of images (the secret and the

appropriate cover images) from the cover selection model selection were used as input to the encoder's model. To embed the secret image in the cover image, just the luminance portion of the image was used by first converting the image to grey scale and then extracting the luminance characteristic of the image. The secret image was converted into its bit equivalence. This was achieved by using Eq. (13):

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.226 & 0.421 & 0.073 \\ -0.128 & -0.181 & 0.221 \\ 0.221 & -0.246 & -0.054 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 64 \\ 286 \\ 286 \end{bmatrix} \quad (13)$$

where, Y represents the luminance feature of the cover image, C_b and C_r represent the blue and red pixel colors respectively. It was discovered that the bit representation of the secret image was very close to the values of the redundant bits in the luminance region (Y) of the cover image. To compensate for this inadequacy, the redundant bits for the luminance feature was multiplied with a constant numeric value. Then, the secret image bits were embedded into the Y region of the cover image, leaving theC_b and C_r untampered. This helped to increase the invisibility property of the stego-image. Although, C_b and C_r bits were adjusted during the training of the encoder model, but at a very insignificant value.

Algorithm
Input: $\text{Secret}_{\text{Image}}$, $\text{Cover}_{\text{Image}}$
Output: $\text{Stego}_{\text{Image}}$
Convert $\text{Secret}_{\text{Image}}$ => $\text{Secret}_{\text{bits}}$
Transform $\text{Cover}_{\text{Image}}$ => Y, C_b, and C_r =>$\text{TCover}_{\text{Image}}$
Scale up Y bits => $\text{Y*Numeric}_{\text{constant}}$
Embeding$_{\text{function}}$($\text{Secret}_{\text{Image}}$, $\text{TCover}_{\text{Image}}$) => $\text{Stego}_{\text{Image}}$
GANdescriminator($\text{Stego}_{\text{Image}}$, $\text{Cover}_{\text{Image}}$):
for i ← 1 to max(noEpoch) **do**:
if($\text{Stego}_{\text{Image}}$!= $\text{Cover}_{\text{Image}}$) **Then**
repeat => GANdescriminator($\text{Stego}_{\text{Image}}$, $\text{Cover}_{\text{Image}}$):
Else
return output as $\text{Stego}_{\text{Image}}$

4.4 Encoder evaluation criteria

The CycleGAN and DCGAN encoder models have been developed at this stage, and the models are used for generating stego-images. In other to measure the payload capacity of the steganography model, Eq. (14) was used. The security of the steganography model was measure using Eqns. (15) and (16) and the security is measured in Decibel (dB) using PSNR. The higher the value of PSNR, the better the security of the system. Also, SSIM can be used to measure the security of the steganography encoding model. The higher the value of SSIM, the better the security of the steganography model.

The payload capacity is measure in bit per pixel.

$$BPP = \frac{Number\ of\ secret\ bits\ embeded}{Total\ pixel\ on\ the\ cover\ image} \quad (14)$$

$$PSNR = 20log_{10}(Max^2C) - 20log_{10}(MSE_{C',C}) \quad (15)$$

$$SSIM(c,c') = [l(c,c')^{\alpha}] . [C(c,c')^{\beta}] . [S(c,c')^{\gamma}] \quad (16)$$

where, BPP=Bit Per Pixel, *PSNR*=Peak Signal-to-Noise Ratio; *SSIM*=Structural Similarity Index Metric.

PSNR and SSIM were the metrics used to measure the imperceptibility assessment of the stego-image when compared with the original cover image, which measures the security of the steganography encoding model. These metrics were discussed in details in chapter three. The SSIM metric was used in case the Mean Squared Error value used in the PSNR metric is equal to zero or less than zero. That is, SSIM becomes relevant when MSE≤0; and the PSNR becomes irrelevant when MSE≤0.

5. RESULT AND DISCUSSION

Tkinter was used to build an interactive user interface for the Hybrid Deep Learning Steganography encoder model. Tkinter is a Python class library for machine learning that provides a graphical user interface. Figure 4 shows the encoding phase user interface, which was developed by using the Tkinter python class library. The interface provides two browser buttons. The first browser button enables the selection of cover image from the cover image directory; the second browse button enables the selection of secret image that the user will like to encode. The interface displays both the cover and the secret image to be encoded.

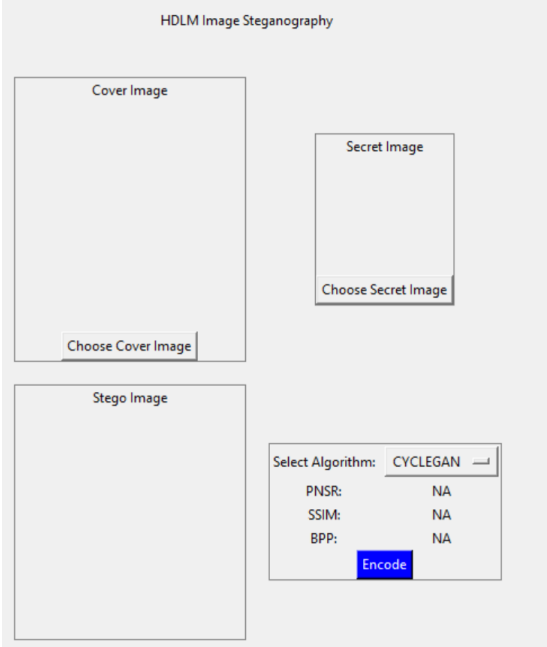

Figure 4. The HDML encoder phase interface

The user interface also contains an event driven button, which commands the encoding process once it is clicked. Before the encoding process, the user makes choice of the encoding model to use. This study allows a user to choose between the cycleGAN encoder model or the DCGAN encoder model. Once a model is selected, the encode button is clicked for the encoding process to begin. Each encoding model presents a stego-image with different evaluation metric values. Although, the two models have good security features, but the cycleGAN encoding model tends to have better metrics values as shown in Figure 5 than the DCGAN encoding model in Figure 6. These results are as tabulated in Table 3.

Table 3 compares the cycleGAN and DCGAN encoding models based on the values of these metrics, which later influences the selection of our proposed cycleGAN model.

Table 3 shows that cycleGAN and DCGAN are good image steganography encoding model. Although, one appears to outperform the other. The proposed cycleGAN encoding

model has a better PSNR value with a better payload capacity and structural similarity index measure value.

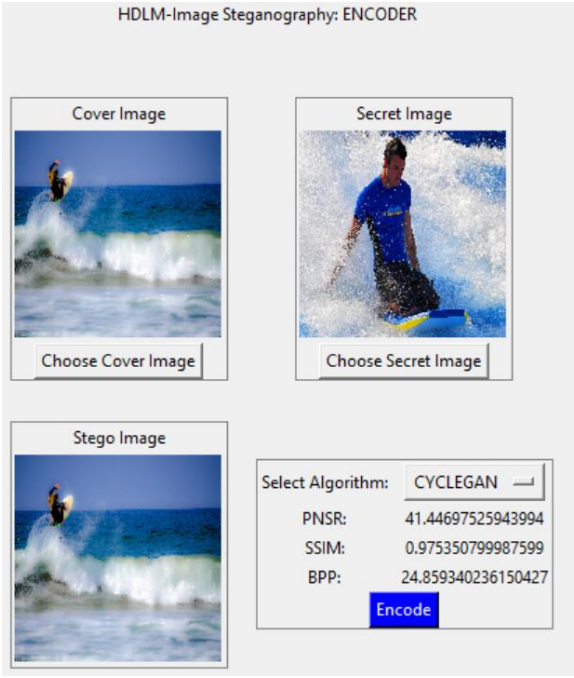

Figure 5. CycleGAN Encoder output metrics

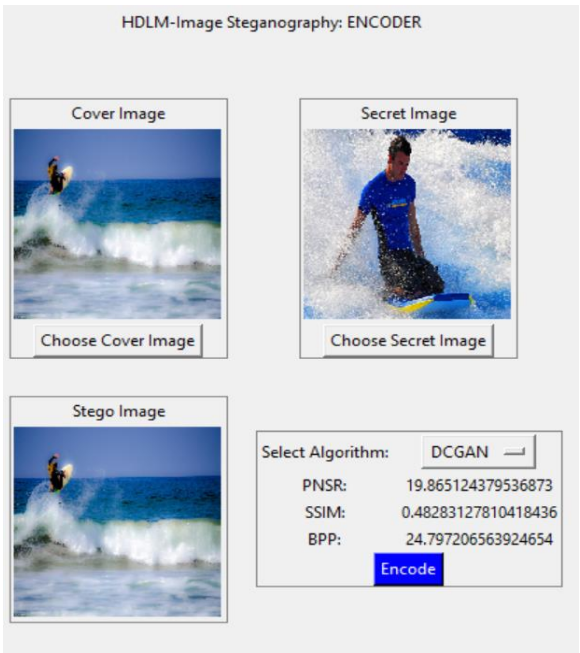

Figure 6. DCGAN Encoder output metrics

Table 3. Encoding models comparison using BPP, PSNR and SSIM

Encoder Model	BPP	PSNR (Db)	SSIM
DCGAN	24.79	19.86	0.48
CycleGAN	24.97	41.45	0.97

6. CONCLUSIONS

We have been able to show that CycleGAN and DCGAN are bothe GAN-based encoding models for image steganography. This study shows that the cycleGAN obtained a better metric value for the payload capacity, PSNR, and SSIM than the DCGAN. The cycleGAN was adopted as the encoding model used in developing the hybrid deep learning model for reversibke image steganography. As stated earlier, this study is an excerpt from a PhD thesis. We hope to complete this study and be able to benchmark the proposed hybrid deep learning model against state-of-the-art models.

REFERENCES

[1] Zhang, C., Benz, P., Lin, C., Karjauv, A., Wu, J., Kweon, I.S. (2021). A survey on universal adversarial attack. International Joint Conferences on Artificial Intelligence (IJCAI) 2021, survey track, arXiv:2103.01498. https://doi.org/10.48550/arXiv.2103.01498

[2] Jiang, R., Zhou, H., Zhang, W., Yu, N. (2017). Reversible data hiding in encrypted three-dimensional mesh models. In IEEE Transactions on Multimedia, 20(1): 55-67. https://doi.org/10.1109/TMM.2017.2723244

[3] Zheng, S., Wang, L., Ling, B., Hu, D. (2017). Coverless information hiding based on robust image hashing. In International Conference on Intelligent Computing, pp. 536-547. https://doi.org/10.1007/978-3-319-63315-2_47

[4] Papadopoulos, N.A., Psannis, K.E. (2018). Sequential multiple LSB methods and real-time data hiding: variations for visual cryptography ciphers. Journal of Real-Time Image Processing, 14(1): 75-86. https://doi.org/10.1007/s11554-016-0630-y

[5] Yi, S., Zhou, Y., Hua, Z. (2018). Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. Signal Processing: Image Communication, 64: 78-88. https://doi.org/10.1016/j.image.2018.03.001

[6] Hureib, E.S., Gutub, A.A. (2020). Enhancing medical data security via combining elliptic curve cryptography and image steganography. International Journal of Computer Science and Network Security, 20(8): 1-8. https://doi.org/10.22937/IJCSNS.2020.20.12.26

[7] Ansari, A.S., Mohammadi, M.S., Parvez, M.T. (2019). A comparative study of recent steganography techniques for multiple image formats. International Journal of Computer Network and Information Security, 11(1): 11-25. https://doi.org/10.5815/ijcnis.2019.01.0

[8] Din, R., Qasim, A.J. (2019). Steganography analysis techniques applied to audio and image files. Bulletin of Electrical Engineering and Informatics, 8(4): 1297-1302. https://doi.org/10.11591/eei.v8i4.1626

[9] Douglas, M., Bailey, K., Leeney, M., Curran, K. (2018). An overview of steganography techniques applied to the protection of biometric data. Multimedia Tools and Applications, 77(13): 17333-17373. https://doi.org/10.1007/s11042-017-5308-3

[10] Kaur, R., Singh, B. (2012). Survey and analysis of various steganographic techniques. International Journal of Engineering Science and Advanced Technology, 2: 561-566. https://doi.org/10.1.1.300.3295

[11] Miri, A., Faez, K. (2018). An image steganography method based on integer wavelet transform. Multimedia Tools and Applications, 77(11): 13133-13144. https://doi.org/10.1007/s11042-017-4935-z

[12] Odeh, A., Elleithy, K., Faezipour, M. (2014). Steganography in text by using MS word symbols. In Proceedings of the 2014 Zone 1 Conference of the American Society for Engineering Education, pp. 1-5. https://doi.org/10.1109/ASEEZone1.2014.6820635
[13] Bailey, K., Curran, K. (2006). An evaluation of image based steganography methods. Multimedia Tools and Applications, 30(1): 55-88. https://doi.org/10.1007/s11042-006-0008-4
[14] Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3): 727-752. https://doi.org/10.1016/j.sigpro.2009.08.010
[15] Chaumont, M. (2020). Deep learning in steganography and steganalysis. In Digital Media Steganography, pp. 321-349. https://doi.org/10.1016/B978-0-12-819438-6.00022-0
[16] Reinel, T.S., Raul, R.P., Gustavo, I. (2019). Deep learning applied to steganalysis of digital images: A systematic review. IEEE Access, 7: 68970-68990. https://doi.org/10.1109/ACCESS.2019.2918086
[17] Baluja, S. (2019). Hiding images within images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(7): 1685-1697. https://doi.org/10.1109/TPAMI.2019.2901877
[18] Zhang, K.A., Cuesta-Infante, A., Xu, L., Veeramachaneni, K. (2019). SteganoGAN: High capacity image steganography with GANs. arXiv preprint arXiv:1901.03892. https://doi.org/10.48550/arXiv.1901.03892
[19] Subramanian, N., Cheheb, I., Elharrouss, O., Al-Maadeed, S., Bouridane, A. (2021). End-to-end image steganography using deep convolutional autoencoders. IEEE Access, 9: 135585-135593. https://doi.org/10.1109/ACCESS.2021.3113953
[20] Wang, Z., Gao, N., Wang, X., Xiang, J., Zha, D., Li, L. (2019). HidingGAN: High capacity information hiding with generative adversarial network. In Computer Graphics Forum, 38(7): 393-401. https://doi.org/10.1111/cgf.13846
[21] Zhang, Z., Fu, G., Ni, R., Liu, J., Yang, X. (2020). A generative method for steganography by cover synthesis with auxiliary semantics. Tsinghua Science and Technology, 25(4): 516-527. https://doi.org/10.26599/TST.2019.9010027
[22] Hu, D., Wang, L., Jiang, W., Zheng, S., Li, B. (2018). A novel image steganography method via deep convolutional generative adversarial networks. IEEE Access, 6: 38303–38314. https://doi.org/10.1109/ACCESS.2018.2852771
[23] Tang, W., Li, B., Tan, S., Barni, M., Huang, J. (2019). CNN-based adversarial embedding for image steganography. IEEE Transactions on Information Forensics and Security, 14(8): 2074-2087. https://doi.org/10.1109/TIFS.2019.2891237
[24] Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., Qin, C. (2019). Reversible image steganography scheme based on a U-net structure. IEEE Access, 7: 9314–9323. https://doi.org/10.1109/ACCESS.2019.2891247
[25] Zhang, Z., Fu, G., Di, F., Li, C., Liu, J. (2019). Generative reversible data hiding by image to-image translation via GANs. Security and Communication Networks, 2019: 4932782. https://doi.org/10.1155/2019/4932782
[26] Duan, X., Guo, D., Liu, N., Li, B., Gou, M., Qin, C. (2020). A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network. IEEE Access, 8: 25777-25788. https://doi.org/10.1109/ACCESS.2020.2971528
[27] Ray, B., Mukhopadhyay, S., Hossain, S., Ghosal, S.K., Sarkar, R. (2021). Image steganography using deep learning-based edge detection. Multimedia Tools and Applications, 80(24): 33475-33503. https://doi.org/10.1007/s11042-021-11177-4
[28] Kadhim, I.J., Premaratne, P., Vial, P.J. (2020). High capacity adaptive image steganography with cover region selection using dual-tree complex wavelet transform. Cognitive Systems Research, 60: 20-32. https://doi.org/10.1016/j.cogsys.2019.11.002
[29] Byrnes, O., La, W., Wang, H., Ma, C., Xue, M., Wu, Q. (2021). Data hiding with deep learning: A survey unifying digital watermarking and steganography. arXiv preprint arXiv:2107.09287. https://doi.org/10.48550/arXiv.2107.09287
[30] Liu, J., Ke, Y., Zhang, Z., Lei, Y., Li, J., Zhang, M., Yang, X. (2020). Recent advances of image steganography with generative adversarial networks. IEEE Access, 8: 60575–60597. https://doi.org/10.1109/ACCESS.2020.2983175.
[31] Chen, B., Wang, J., Chen, Y., Jin, Z., Shim, H.J., Shi, Y. (2020). High-capacity robust image steganography via adversarial network. KSII Transactions on Internet and Information Systems, 14(1): 366–381. https://doi.org/10.3837/tiis.2020.01.020
[32] Petzka, H., Fischer, A., Lukovnicov, D. (2017). On the regularization of Wasserstein GANs. arXiv preprint arXiv:1709.08894. https://doi.org/10.48550/arXiv.1709.08894
[33] Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. https://doi.org/10.48550/arXiv.1411.1784
[34] Borji, A. (2022). Pros and cons of GAN evaluation measures: New developments. Computer Vision and Image Understanding, 215(C). https://doi.org/10.1016/j.cviu.2021.103329