International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Deep Convolutional Generative Adversarial Network for Image Steganography Enhancement

Syeda Imrana Fatima*[1], Yugandhar Garapati[2]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: In recent times, safeguarding data has emerged as a paramount global issue demanding the utmost attention and concern. The secret data is exposed to potential hacks when transmitted via conventional communication channels. The image steganalysis development based on the Convolutional Neural Network (CNN) has become challenging for image steganography. However, the recent steganographic approaches are complex to resist the detection of CNN-based steganalyzers. To address this issue, this research proposed the image steganographic plan based on a Deep Convolutional Generative Adversarial network (DCGAN) with adversarial attack. The proposed method utilized the MNIST steganography dataset to estimate the performance of DCGAN. This is performed to generate the secure DCGAN result, which has greater robustness to adversarial data operations. The experimental results show that the proposed method achieves greater performance and achieves the stego accuracy of 0.9155, discriminator loss of 0.0307 as well as similitude loss of 0.00167 when compared to the existing methods like GAN and Information-driven GAN (IDGAN). The proposed approach can efficiently protect the sensitive data even affecting the quality of image data as well as outperforms the compared existing methods.

Keywords: *Adversarial Attacks, Convolutional Neural Network, Deep Convolutional Generative Adversarial Network, Discriminator and Image Steganography*

1. Introduction

Generative Adversarial Network (GAN) is the most significant development in the Deep Learning (DL) domain, particularly for image applications [1, 2]. GAN consists of two neural networks: the generator can collect the distribution of data and the discriminator that predicts the sample composition from training data. The generator can learn from the discriminator feedback and does not have access to the real data. The GAN is used to reduce the data imbalance between the generator and discriminator by partitioning the training data based on the data label [3, 4]. Other variants of GAN, the conditional GAN (CGAN) utilize the class label during training as an input to both generator and discriminator [5, 6]. Using a generator for each class, data was split according to label and processed each class independently. Each factor relates to a unique data class through the data division. According to the labels, a data splitter was performed and removed the variance between classes and the role was to reconstruct the joint data distribution for combining the image data with class labels [7].

Various researchers have introduced steganalysis approaches for securing image data from hackers. The embedding process in the steganography is utilized to hide the secret data in an image, video, audio, and message [8, 9]. Likely, some researchers have introduced the steganography without embedding approach, where the message is embedded by noise vector tools as well as a generator [10, 11]. The image steganography approach utilizes the powerful learning capability of neural networks to help identify the most appropriate embedding positions in a cover image. The high-resolution image distributes the suitable place to develop the new data, modifying the image without a user experience [12, 13]. In this situation, the Convolutional Neural Network (CNN) has been utilized to recover the image data from hidden data. However, CNN is vulnerable to adversarial attacks and weakens the robustness of a model. In image classification approaches, the robustness can be enhanced by extending small modifications to an input image based on inciting an error in the classification approach [14, 15]. This research proposes the Deep Convolutional GAN (DCGAN) approach to enhancing image steganography. The major contributions involved in this research are given as follows:

- The DCGAN approach is capable of performing adversarial attacks over the steganalysis approach of existing methods.
- DCGAN compromises the discriminator is regarded as the leader and the generator is regarded as the follower.
- In terms of its anti-analysis ability, the DCGAN model greatly exceeded traditional image steganography approaches using the information-driven approach for producing covert images.

[1] *Department of Computer Science & Engineering, GITAM (Deemed to be University), Hyderabad, India*
[2] *Department of Computer Science & Engineering, GITAM (Deemed to be University), Hyderabad, India*
ORCID ID: 0000-0003-2708-9714
** Corresponding Author Email: isyeda@gitam.in*

This paper is arranged as follows: Section 2 provides the literature survey. Section 3 presents the proposed methodology. Section 4 provides the results and discussion. The conclusion of this research paper is given in Section 5.

2. Literature Survey

Alejandro Martin et al. [16] introduced the GAN to enhance the capability of the spatial domain steganalysis approach as well as applied the personal data with less image modification. Through the training process, GAN utilized the Least Significant Bit (LSB) steganography approach for learning a message to adapt an image. The outcomes indicated that an approach was successful at avoiding detection through the existing DL steganalysis architecture. However, the suggested approach needed minimal computational resources due to the complexity of the algorithm.

Fei Peng et al. [17] presented the image steganography architecture based on the GAN generator as well as gradient descent calculation. During data embedding, the secret data was primarily plotted into stego noise through particular mapping instruction, and it was utilized as input to the GAN generator to generate a stego image. A data extraction was accomplished by iteratively updating a noise vector by a gradient descent generator. Eventually, secret data was extracted from the updated noise vector. A suggested approach has better generalization with various GAN approaches as well as image datasets. However, the secret data extraction had a problem in stego process due to an irreversibility generator.

Chunying Zhang et al. [18] presented the steganography approach according to the new Information-driven GAN (IDGAN), which fused the GAN, Attention Mechanisms as well as image interpolation approaches. The attention mechanism was developed on top of the actual GAN approach to enhance the accuracy of an image. In the generation model, the GAN replaced some transposed convolution operations with image interpolation for better the quality of dense images. The IDGAN generated the images that involved secret data without utilizing the cover images and the GAN for data embedding. The IDGAN utilized an attention mechanism to enhance the image information as well as optimize the steganography effect by an image interpolation approach. However, the suggested approach scrap to maintain particular image types like difficult medical images.

Magdy M. Fadel et al. [19] developed a framework for hiding the secret data in a spatial domain through the partition of the host image into non-overlapping blocks. The Whale Optimization Algorithm (WOA) was used for the classification of every non-overlapping block as edges and smooth. Various WOA objective functions could be used to identify every pixel embedding size according to intensity. The smooth and edge blocks were utilized for the fitness as well as cost function during the embedding of the data. Furthermore, various groupings of skimming levels as well as beginning opinions for every block in the host image were identified to minimize embedding misrepresentation. However, the number of possibility estimations as well as model interpretations influences the computational time.

Linna Wang et al. [20] presented the dynamic watermarking method according to a reversible image-hiding network, which enhanced DNN watermark unpredictability as well as it could efficiently redevelop the secret image of the DNN approach. The suggested approach utilized the MNIST, fashion-MNIST, CIFAR-10, CIFAR-100 as well as Caltech-101 datasets. The suggested approach obtained the maximum DNN watermarking accuracy as well as maximum unpredictability with no side effects on significant functions of the host DNN approach. However, the suggested approach acquired the less performance results by utilizing of MNIST dataset.

Ehsan Nazari et al. [21] implemented a systematic approach called Auto GAN, which automated the training process involved in quantitative measures. The Auto GAN algorithm not only determined the better stopping point for training the GAN but also allowed one to run some training models for achieving better performance with GAN outcomes. There was a limited responsible period to review the number of images and visual investigation methods were constrained to images and tabular data.

Chao Yuan et al. [22] introduced the end-to-end image steganography plan according to GAN with adversarial attack as well as pixel-wise deep fusion. The universal adversarial network was used in the attack module for CNN-based steganalysis for the enhancement of security. The encoder model was utilized as a generator to implement the pixel-wise deep fusion for invisible data embedded with a high payload. The decoder model was responsible for recovering the embedded data procedure. A critic module was developed for discriminators to provide objective scores as well as conduct adversarial training. However, the suggested approach had factors such as dimension as well as quantity of latent vectors that could significantly influence the performance results.

By analyzing the various state-of-the-art methods for image steganography, some limitations have been identified: computational complexity, generator irreversibility, and poor performance due to limited sample data. The number of possibility estimations as well as model interpretations influences the computational time. The factors such as dimension as well as quantity of latent vectors could significantly influence the performance results. To overcome these challenges, this research proposed the DCGAN approach for the enhancement of image steganography security.

3. Proposed Methodology

In this research, a DCGAN is proposed to enhance image steganography, a learning procedure follows a group of changes which only functional for greater robust stenographic approach. GAN performs the input image to further develop a steganographic message. Fig. 1 depicts the workflow of the suggested method.

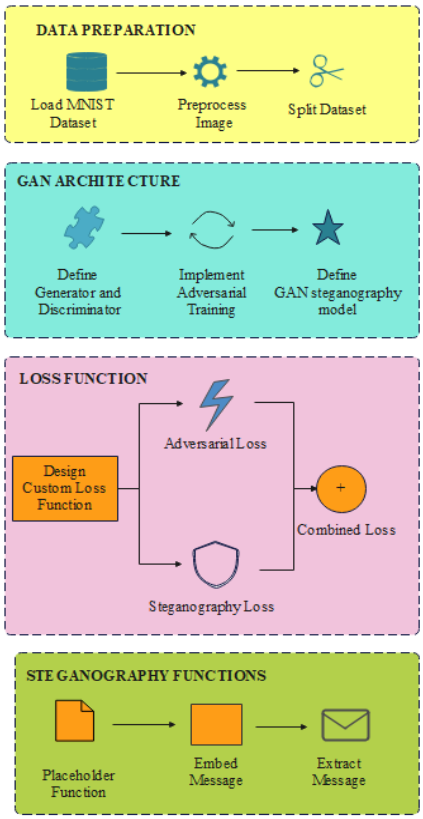

Fig. 1. Block diagram of the proposed method

3.1. Basic Principle

The principle of developing an image steganography scheme is that can embed the secret data on the premise of satisfaction of security and imperceptibility. To encounter these requirements, this research implements the DCGAN as the basic framework of image steganography as well as designs various modules to achieve certain goals. The entire steganography approach is the collection of sub-networks such as encoder, decoder, and discriminator. The encoder can hide the secret image from a cover image with a similar size. The steganography images are generated through the encoder, and after that secret images are extracted by the decoder. The discriminator takes the cover and steganography images as input to identify whether the input image contains secret images. Fig. 2 shows the workflow of the system architecture.

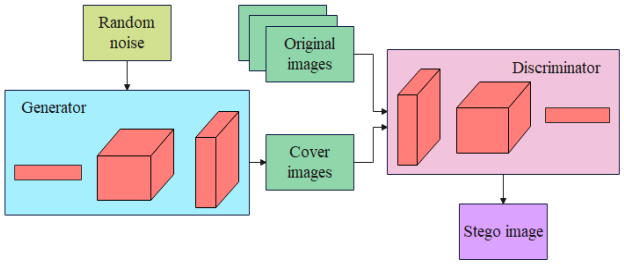

Fig. 2. Workflow of the system architecture diagram

To achieve high security, a cover and stego image should not be efficiently classified through Steg analyzers. Steganalysis examines whether an image contains secret data by utilizing the benefits of statistical characteristic variation among cover and stego images, thus the proposed method distracts the difference to mislead the steganalyzer with adversarial attack. For imperceptibility, the distortion of the cover image is difficult to perceive and the variation among cover as well as stego images are estimated. If the variation is trivial, the image can be considered as high quality. In addition, there are various areas in the spatial domain with edges and complex textures, these areas represent high-frequency image parts. Based on the steganography concept, the detection of stego images is complex, because of the modification of the high-frequency image parts. Hence, GCGAN aims to hide secret data in peripheral as well as complex textured areas to improve imperceptibility as well as security. Fig. 3 depicts the hyperparameter tuning model.

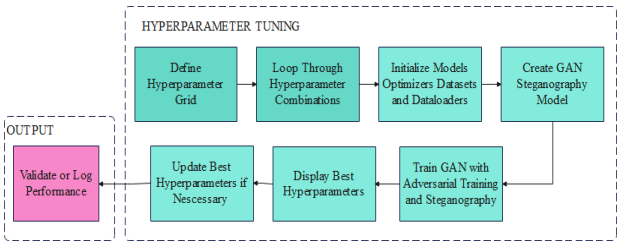

Fig. 3. Process of hyperparameter tuning

3.2. Dataset and pre-processing

At the initial stage of this research, the dataset is collected to estimate the performance of the proposed method. The proposed method used the benchmark dataset of MNIST to train and test the DL approaches. The MNIST is a larger handwritten digital dataset that involves 70,000 gray images as well as a test set of 10,000 images. Every image has a size of 28 × 28 pixels and its class label ranges from between 0 and 9. In the pre-processing step, the hamming code can be utilized to encode the data as well as plot the encoded binary data randomly, which enhances the data confidentiality. Furthermore, normalizing the plotted data can be supported to mitigate the vanishing problems as well as exploring the gradients during training. This pre-processing step distributes the effective model as well as reliable input data, in that way, it enhances the performance of the model.

3.3. Deep Convolutional Generative Adversarial Network

The GAN involves two general networks such as generator (G) and discriminator (D). The generator makes use of random noise to produce fake samples, which are then forwarded to the discriminator for the determination of combined with actual samples. A discriminator is required to differentiate whether an input model is true or false. This training process aims to attain Nash equilibrium as well as generator can generate the models with similar distribution. The GAN-based optimization is to reduce generator and discriminator losses. This means an identified value from the discriminator for actual and generated samples is 0.5. The training process is explained in Eq. (1) as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim Pdata(x)}\left[\log\left(D(x)\right)\right] + E_{z\sim P_{z(z)}}\left[\log\left(1-D\left(G(z)\right)\right)\right] \quad (1)$$

Where, $Pdata$ – distribution of actual samples; P_z – prior input noise distribution, $G(z)$ – generated model through generator as well as $D(.)$ – identified value from discriminator.

DCGAN is an irregular approach to GAN, which adopts convolution and transposed convolutional layers to exchange initial fully connected layers in actual GAN. In addition, ReLU is adopted as an activation function of the generator network as well as Leaky ReLU is utilized as an activation function of the discriminator network. The DCGAN is majorly utilized for various instances because of its enhanced training stability and high quality of generated samples. Even though the DCGAN helps acquire the stable, unstable training problem still happens frequently, which originates the mode failure as well as gradient disappearance. Furthermore, spectral normalization is the new weight normalization approach which attains a Lipschitz limitation by confining weight matrix *L2* spectral normalization of every layer and improving the flexibility of DCGAN. Hence, the DCGAN is developed by introducing spectral normalization, and Wasserstein distance with Gradient Penalty (GP) to improve training stability as well as generate satisfying models. The loss function of the DCGAN is formulated in Eq. (2) and (3) as follows:

$$Loss = E[D(y)] - E[D(x)] + \mu E[(1 - \|H_z D(z)\|_2)^2] \quad (2)$$

$$\|H_z D(z)\| = \frac{|D(z_1)-D(z_2)|}{\|z_1-z_2\|} \quad (3)$$

Where, $D(x)$, $D(y)$ – corresponding predicted value from discriminator of actual as well as generated sample. μ – weight constraint coefficient; P_z – a joint distribution of generated as well as actual models. While the Wasserstein distance, as well as GP, are presented in DCGAN, a gradient tends to be close to a particular stable value during the training process. Furthermore, the Lipschitz constraint condition is further guaranteed by developing the spectral normalization. These two schemes contribute to justifying vanishing gradient as well as exploding gradient problems.

Since the convolutional kernel of an approach utilizing DCGAN is grouped, it cannot collect the global data. The self-attention mechanism can extract the features by establishing the relationships among local and isolated regions. Hence, the performance of the generated approach and developed sample quality can be additionally enhanced by generating the self-attention mechanism. The spectral normalization is arranged after every convolutional as well as deconvolutional layer of both the generator and discriminator instead of actual batch normalization. In addition, self-attention approaches are extended after the final two layers of both the generator as well as discriminator.

3.4. Loss Function

In the proposed method, total loss L_{total} is determined to train an encoder E, generator G, structure encoding generator G_{STRU} as well as extractor Ex, which is expressed in Eq. (4) as follows:

$$L_{total} = L_E + L_G + \alpha_{Ex} + L_{Ex} \quad (4)$$

Where, α_{Ex} permits the balance between the synthesis quality as well as extraction accuracy. In this proposed method, α_{Ex} is 15 to guarantee the successful principal component hiding secret messages. The encoding loss L_E is acquired through associating the encoder distance loss $L_{E,dist}$ as well as encoder structure loss $L_{E,stru}$. The L_E is formulated in Eq. (5) as follows.

$$L_E = L_{E,dist} + L_{E,stru} \quad (5)$$

Where, D_{DIST} is used to ensure that T_1 confirms to a uniform distribution $U(-1,1)$. Where $L_{E,stru} = \left|\hat{S}_2 - \hat{S}_1\right|^1.\hat{S}_i$ denotes the structural feature of the redeveloped image. The L_G is generated by combining the redeveloped loss $L_{G,rec}$, texture loss $L_{G,texture}$ as well as adversarial loss $L_{G,real}$. L_G is expressed in Eq. (6) as follows:

$$L_G = L_{G,rec} + L_{G,texture} + 2 \times L_{G,real} \quad (6)$$

Where, the higher weight for $L_{G,real}$ guarantees the development quality. $L_{G,rec}$ is estimated by utilizing the $L1$ loss between actual image I as well as reconstructing the image $\hat{I}_1$. $L_{G,texture}$ is estimated by utilizing the developing $\hat{I}_2$ with similar texture feature as image I, but a different structure S_2. Then, the features are provided with the arbitrarily cropped patches of I as well as $\hat{I}_2$ to the co-occurrence discriminator D_{co}. $L_{G,real}$ is developed to design all synthesized images $\hat{I}_1$, $\hat{I}_2$ and $\hat{I}_3$ from actual images, which is expressed in Eq. (7) as follows:

$$L_{G,real} = D(\hat{I}_1) + D(\hat{I}_2) + D(\hat{I}_3) \quad (7)$$

The tensor extracting loss, L_{Ex} is estimated by $L1$ loss is expressed in Eq. (8) as follows:

$$L_{Ex} = |\hat{Z} - Z|^1 \quad (8)$$

Where, $\hat{Z}$ –extracted secret tensor from E_x, Z – secret tensor.

3.5. Steganography Function

In this section, the placeholder function is provided to the embedding process and then embedded data is forwarded to the extraction process. The detailed description of this function is explained in the following section.

3.5.1. Embedding

After that key models are acquired by hiding blocks $R_{conceal}$, the further step is to embed the DCGAN in an actual host network. The key samples X_{key} as well as normal sample X are split each other as the novel training set of an actual host network, after forwarded to the actual host network O to acquire the DCGAN network, which is expressed in Eq. (9) as follows:

$$Train(O, X, X_{key}) \rightarrow DCGAN \quad (9)$$

The DCGAN network not only contains the hash functions of usual samples but also has key samples, which are expressed in Eq. (10) as follows:

$$Test(DCGAN, X) \rightarrow Y, Test(DCGAN, X_{key}) \rightarrow Y_{key} \quad (10)$$

Due to the parameters of an actual host network being jobless, the key models will not cause the performance of the actual host network in the usual sample classification.

3.5.2. Data Extraction

After the receiver acquires $stego$, the secret data is extracted by the following steps:

Step 1: Initially, the pre-processing is performed by the $stego$ as well as matrix $target$ whose values are in between -1 and 1 is acquired. The pre-processing procedure is given in Eq. (11) as follows:

$$target = \frac{2 \times stego}{255} - 1 \quad (11)$$

Step 2: Solve an equation $target = G(z)$. Initially, the parameter set of G as well as the gradient descent approach are utilized to iteratively update the parameters of vector z. An error function is formulated in Eq. (12) as follows:

$$\min_z Error = MSE[G(z) - target] \quad (12)$$

Where, $MSE(.)$ – mean square error function. Specifically, once $z = z_{stego}$, there is $target = g(z)$ as well as $Error = 0$.

Step 3: A vector is arbitrarily sampled from a standard normal distribution and it is performed as initial value of z which is input to the generator as well as iteratively updated. The iteration rule of vector z is expressed in Eq. (13) and (14) as:

$$z_{init} = N(0,1) \quad (13)$$

$$z_{ing} \leftarrow \left[z_{ing} - y_z \frac{\partial Error}{\partial z}\right] \quad (14)$$

Where, y_z – step size.

Step 4: When the error drops to a particular stage, a generator will give the output similar to $stego$ image, and the noise vector z_{ed} is extremely correlated with $stego$ noise vector z_{stego}, which are expressed in Eq. (15) and (16) as:

$$Error = MSE[G(z_{ed}) - target] \approx 0 \quad (15)$$

$$z_{ed} \approx z_{stego} \quad (16)$$

Step 5: z_{ed} is normalized to obtain z_{nor}, whose range of components are constrained in $[-1, 1]$, which is expressed in Eq. (17) as follows:

$$z_{nor} = \begin{cases} \frac{z_{ed}}{min(z_{ed})}, z_{ed} \leq 0 \\ \frac{z_{ed}}{max(z_{ed})}, z_{ed} > 0 \end{cases} \quad (17)$$

Where, $min(z_{ed})$, $max(z_{ed})$ – minimum as well as maximum value in z_{ed}; z_{nor} – approximate solution of equation $target = G(z)$.

Step: A binary secret data is extracted from z_{nor} based on an inverse mapping rule.

4. Experimental Results

The performance of the proposed DCGAN is simulated on the Python 3.9 environment, windows 10 operating system and Intel i7 processor. The effectiveness of the proposed method is estimated by various steganography approaches. The Structural Similarity (SSIM), as well as the Peak Signal-to-Noise Ratio (PSNR), are assumed for computing the imperceptibility. The mathematical expression of these metrics is expressed in Eq. (18) and (19) as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (18)$$

$$PSNR(x, y) = 101g\left(\frac{MAX_I^2}{MSE_{(x,y)}}\right) \quad (19)$$

Where, μ_x, μ_y – mean image value x and y; σ_x^2, σ_y^2 variance of x and y; σ_{xy} covariance; c_1, c_2 – constants to roughly eliminate the denominator because of much small $\mu_x^2 + \mu_y^2$ as well as $\sigma_x^2 + \sigma_y^2$; MAX_I – maximum possible pixel value of x and y; $MSE_{(x,y)}$ – mean square error from x to y.

4.1. Performance Analysis

To accurately estimate a DCGAN approach, various tests

are employed with GAN architecture. Table 1 and Fig. 4 show the accuracy results of the pre-trained discriminator using the MNIST dataset.

Table 1. Accuracy of pre-trained discriminator

	Accuracy	TPR	TNR
Training Set	0.973	0.975	0.982
Validation Set	0.942	0.954	0.945
Test Set	0.981	0.962	0.936

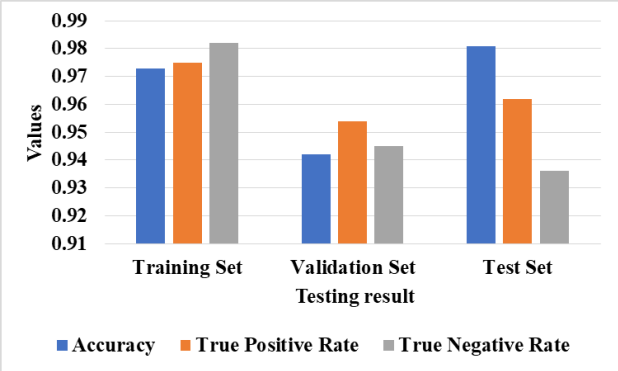

Fig. 4. Graphical representation of pre-trained discriminator model

At the training end, the generated models are estimated. The accuracy, discriminator loss as well as minimizing loss attains better results in the final epoch. But the similitude loss minimizes in further training epochs as well as it retains acceptable values in the entire training process. Table 2 shows the results acquired with DCGAN in the validation set.

Table 2. Results acquired with DCGAN in the validation set

	Accuracy	Discriminator Loss	Similitude loss	Epoch
1	0.007698	0.00568	0.02	3
2	0.003	0.0156	0.003	11
3	0.003	0.0174	0.0037	17
4	0.004	0.0059	0.00417	25
5	0.005	0.0145	0.00197	26
6	0.008	0.027	0.00195	28
7	0.008	0.028	0.00181	30
8	0.012	0.0308	0.00175	32
9	0.012	0.0358	0.00180	34
10	0.015	0.0304	0.00176	35

Table 3 shows the hyper-parameters performed during GA execution. The hyperparameters utilized in GA are as follows: μ – population size at every generation. λ – number of crossovers performed at every epoch.

Table 3. Hyper-parameters performed during GA execution

Hyperparameter	Value	Hyperparameter	Value
Mutation Probability	0.25	Max depth	32
Epochs per individual	2	λ	5
Generations	10	μ	10
New layer probability	0.3	Crossover probability	0.5

Table 4 and Fig. 5 represent the outcomes in the test set of best individuals acquired in every execution of GA. The outcomes are compared to these acquired values with the physically designed generator; however, the acquired values are most effective.

Table 4. Performance of better individual acquired in every execution of GA

Execution	Similitude	Discriminator loss	Stego accuracy	Training time (s)
1	0.003247481	0.00842853	0.000536747	187
2	0.013035204	0.022285271	0.012498213	187
3	0.003589422	0.008731116	0.000315445	118
4	0.003254174	0.010199631	0.000789523	71
5	0.003296 52	0.010699377	0.00284327	63

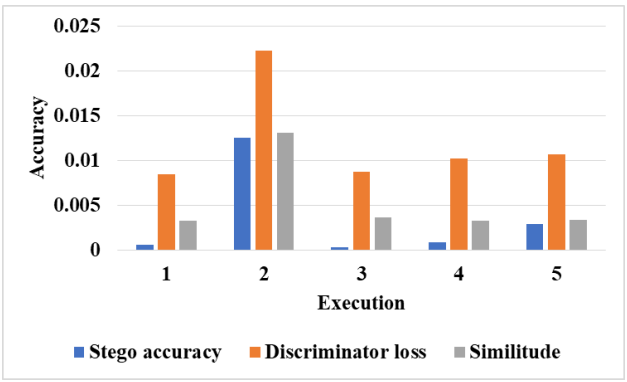

Fig. 5. Graphical representation of accuracy results with the number of executions

Table 5 illustrates the 10 execution results of GA retraining the discriminator as well as the outcomes of the pre-trained modal. In every execution stage, the ending discriminator can classify an actual test set exceeding the discriminator.

Eventually, the outcomes in the execution process are unbalanced. This section shows how the suggested approach follows the balance among cover as well as stego accuracy, which originates greater falls at a few points. This is because of the over-fitting of one class, causing a greater number of misclassifications compared to other methods.

Table 5. Pre-trained discriminator estimated utilizing test set

Execution	Cover accuracy	Global accuracy	Stego accuracy
Pre-Trained YeNet	0.92	0.918	0.916
1	0.89675	0.92375	0.95075
2	0.947	0.92875	0.9105
3	0.9135	0.931375	0.94925
4	0.93875	0.928375	0.918
5	0.9175	0.929625	0.94175
6	0.91675	0.925125	0.9335
7	0.90275	0.931125	0.9595
8	0.9305	0.930875	0.93125
9	0.93725	0.92325	0.90925
10	0.9345	0.925	0.9155

Table 6 shows the DCGAN results estimated utilizing the test set. A similitude loss is much near to Pareto Front as well and misclassification obtains more than 99% with stego fake images. Hence, it is applicable to authorize that the last mode attains anticipated goals. The DCGAN achieves the accuracy of 0.008 respectively.

Table 6. DCGAN results estimated utilizing the test set

Accuracy	Discriminator loss	Similitude loss
0.008	0.03075876	0.00167084

4.3 Comparative Analysis

Table 7 represents the comparative analysis of the proposed method. The proposed method's performance is evaluated by utilizing evaluation metrics like accuracy, detection rate, PSNR, SSIM as well as discriminator loss.

Table 7. Comparative analysis of the proposed method

Methods	Accuracy (%)	Detection rate (%)	PSNR	SSIM	Discriminator loss
GAN [16]	N/A	N/A	N/A	N/A	0.0106
IDGAN [18]	97.2	10.8	N/A	N/A	N/A
DNN [20]	91.61	N/A	33.74	0.9895	N/A
GAN-CNN [22]	N/A	N/A	43.7526	0.9884	N/A
Proposed DCGAN	98.3	26.5	45.3736	0.9935	0.0023

4.3 Discussion

In this section, the advantages of the proposed method and the limitations of existing methods are discussed. The existing method has some limitations such as the GAN [16] required minimal computational resources because of algorithm complexity. IDGAN [18] had scrap to maintain particular image types like difficult medical images. The DNN [20] had acquired less performance results by utilizing the MNIST dataset. The GAN-CNN [22] had factors such as dimension as well as quantity of latent vectors that could significantly affect the performance. The proposed DCGAN-based image steganography approach outperforms the limitations of these existing methods. The DCGAN has a higher classification accuracy than the GAN. However, the GAN can be vulnerable to attacks that extract sensitive data from the training data. The proposed method achieves an accuracy of 98.3%, a detection rate of 26.5%, a PSNR of 45.3736, SSIM of 0.9935 as well a discriminator loss of 0.0023 respectively.

5. Conclusion

This research proposed the DCGAN based image steganography with adversarial attack. This research utilized the MNIST steganography dataset to estimate the performance of the proposed method. The proposed method supplies the new secret image extraction mechanism by utilizing the DCGAN generator as well as gradient descent estimation, which makes it simpler to extract the DCGAN-based image steganography. This research establishes the adversarial attack potentials performed in image steganography for enhancing the security of image data. The experimental results as well as the implementation of the proposed methods represent better performance in data extraction, message embedding, hiding the data efficiently

as well as robustness in quality estimation metrices. In the future, the proposed method will extend to utilizing the various DL approaches to enhance the accuracy as well as robustness of the data hiding.

Author contributions

Syeda Imrana Fatima: Conceptualization, Methodology, Software, Field study, Writing-Original draft preparation, **Yugandhar Garapati:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Y. Chen, Q. Gao, and X. Wang, "Inferential Wasserstein generative adversarial networks," *J. R. Stat. Soc. B*, vol. 84, no. 1, pp. 83-113, Feb. 2022. https://doi.org/10.48550/arXiv.2109.05652

[2] N. A. Mashudi, N. Ahmad, and N. M. Noor, "LiWGAN: A Light Method to Improve the Performance of Generative Adversarial Network," *IEEE Access*, vol. 10, pp. 93155-93167, Aug. 2022. https://doi.org/10.1109/ACCESS.2022.3203065

[3] S. Zhang, K. Huang, Z. Qian, R. Zhang, and A. Hussain, "Improving generative adversarial networks with simple latent distributions," *Neural Comput. Appl.*, vol. 33, pp. 13193-13203, Oct. 2021. https://doi.org/10.1007/s00521-021-05946-3

[4] G. Baykal, F. Ozcelik, and G. Unal, "Exploring deshufflegans in self-supervised generative adversarial networks," *Pattern Recognit*., vol. 122, p. 108244, Feb. 2022. https://doi.org/10.1016/j.patcog.2021.108244

[5] M. Lupo Pasini, V. Gabbi, J. Yin, S. Perotto, and N. Laanait, "Scalable balanced training of conditional generative adversarial neural networks on image data," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 13358-13384, Nov. 2021. https://doi.org/10.48550/arXiv.2102.10485

[6] P. Shamsolmoali, M. Zareapoor, L. Shen, A. H. Sadka, and J. Yang, "Imbalanced data learning by minority class augmentation using capsule adversarial networks," *Neurocomputing*, vol. 459, pp. 481-493, Oct. 2021. https://doi.org/10.48550/arXiv.2004.02182

[7] M. Mohebbi Moghaddam, B. Boroomand, M. Jalali, A. Zareian, A. Daeijavad, M. H. Manshaei, and M. Krunz, "Games of GANs: Game-theoretical models for generative adversarial networks," *Artif. Intell. Rev.*, vol. 56, pp. 9771-9807, Feb. 2023. https://doi.org/10.1007/s10462-023-10395-6

[8] S. Ke and W. Liu, "Consistency of multiagent distributed generative adversarial networks," *IEEE Trans. Cybern*., vol. 52, no. 6, pp. 4886-4896, Oct. 2020. https://doi.org/10.1109/TCYB.2020.3022695

[9] S. Agarwal, C. Kim, and K. H. Jung, "Steganalysis of Context-Aware Image Steganography Techniques Using Convolutional Neural Network," *Appl. Sci*., vol. 12, no. 21, p. 10793, Oct. 2022. https://doi.org/10.3390/app122110793

[10] J. Luo, P. He, J. Liu, H. Wang, C. Wu, C. Yuan, and Q. Xia, "Improving security for image steganography using content-adaptive adversarial perturbations," *Appl. Intell*., vol. 53, no. 12, pp. 16059-16076, Jun. 2023. https://doi.org/10.1007/s10489-022-04321-6

[11] R. Huang, C. Lian, Z. Dai, Z. Li, and Z. Ma, "A novel hybrid image synthesis-mapping framework for steganography without embedding," *IEEE Access*, Oct. 2023. https://doi.org/10.1109/ACCESS.2023.3324050

[12] L. Li, W. Zhang, C. Qin, K. Chen, W. Zhou, and N. Yu, "Adversarial batch image steganography against CNN-based pooled steganalysis," *Signal Process*., vol. 181, p. 107920, Apr. 2021. https://doi.org/10.1016/j.sigpro.2020.107920

[13] A. P. P. Aung, X. Wang, R. Yu, B. An, S. Jayavelu, and X. Li, "DO-GAN: A Double Oracle Framework for Generative Adversarial Networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11275-11284. https://doi.org/10.48550/arXiv.2102.08577

[14] V. Costa, N. Lourenço, J. Correia, and P. Machado, "Improved evolution of generative adversarial networks," in *Proceedings of the genetic and evolutionary computation conference companion*, 2021, pp. 145-146. https://doi.org/10.1145/3449726.3459448

[15] Y. H. Li, C. C. Chang, G. D. Su, K. L. Yang, M. S. Aslam, and Y. Liu, "Coverless image steganography using morphed face recognition based on convolutional neural network," *EURASIP J. Wireless Commun. Networking*, vol. 2022, p. 28, Dec. 2022. https://doi.org/10.1186/s13638-022-02107-5

[16] A. Martín, A. Hernández, M. Alazab, J. Jung, and D. Camacho, "Evolving Generative Adversarial Networks to improve image steganography," *Expert Syst. Appl*., vol. 222, p. 119841, Jul. 2023. https://doi.org/10.1016/j.eswa.2023.119841

[17] F. Peng, G. Chen, and M. Long, "A robust coverless steganography based on generative adversarial networks and gradient descent approximation," *IEEE Trans. Circuits Syst. Video Technol*., vol. 32, no. 9, pp.

5817-5829, Mar. 2022. https://doi.org/10.1109/TCSVT.2022.3161419

[18] C. Zhang, X. Gao, X. Liu, W. Hou, G. Yang, T. Xue, L. Wang, and L. Liu, "IDGAN: Information-Driven Generative Adversarial Network of Coverless Image Steganography," *Electronics*, vol. 12, no. 13, p. 2881, Jun. 2023. https://doi.org/10.3390/electronics12132881

[19] M. M. Fadel, W. Said, E. A. Hagras, and R. Arnous, "A Fast and Low Distortion Image Steganography Framework Based on Nature-Inspired Optimizers," *IEEE Access*, vol. 11, pp. 125768-125789, Oct. 2023. https://doi.org/10.1109/ACCESS.2023.3326709

[20] L. Wang, Y. Song, and D. Xia, "Deep neural network watermarking based on a reversible image hiding network," *Pattern Anal. Appl*., vol. 26, pp. 861-874, Feb. 2023. https://doi.org/10.1007/s10044-023-01140-4

[21] E. Nazari, P. Branco, and G. V. Jourdan, "AutoGAN: An Automated Human-Out-of-the-Loop Approach for Training Generative Adversarial Networks," *Mathematics*, vol. 11, no. 4, p. 977, Feb. 2023. https://doi.org/10.3390/math11040977

[22] C. Yuan, H. Wang, P. He, J. Luo, and B. Li, "GAN-based image steganography for enhancing security via adversarial attack and pixel-wise deep fusion," *Multimedia Tools Appl*., vol. 81, no. 5, pp. 6681-6701, Feb. 2022. https://doi.org/10.1007/s11042-021-11778-z

[23] MNIST dataset link: https://www.kaggle.com/code/manthansolanki/image-classification-with-mnist-dataset.