Received: 17 February 2024 | Revised: 15 June 2024 | Accepted: 27 June 2024

IET Image Processing

DOI: 10.1049/ipr2.13177

ORIGINAL RESEARCH

VidaGAN: Adaptive GAN for image steganography

Vida Yousefi Ramandi | Mansoor Fateh | Mohsen Rezvani

Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran

Correspondence
Mansoor Fateh, Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran.
Email: mansoor_fateh@shahroodut.ac.ir

Abstract

A recent approach to image steganography is to use deep learning. Mainly, convolutional neural networks can extract complex features and use them as patterns to combine hidden messages and images. Also, by using generative adversarial networks, it is possible to generate realistic and high-quality stego images without any noticeable artifacts. Previous methods suffered from challenges such as simple architecture, low network accuracy, imbalance between capacity and transparency, vanishing gradients, and low capacity. This study introduces a steganography framework named VidaGAN that utilizes deep learning techniques. The network being proposed is made up of three components: an encoder, a decoder, and a critic, and introduces a novel architecture and several innovations to address some of the unresolved challenges mentioned above. This study introduces a novel method for embedding any type of binary data into images using generative adversarial networks, enabling us to enhance the visual appeal of images generated by the specified model. This neural network called VarIable aDAptive GAN (VidaGAN) achieved state-of-the-art status by reaching a hiding capacity of 3.9 bits per pixel in the DIV2K dataset. Furthermore, examination by the StegExpose steganalysis tool shows an AUC of 0.6, a suitable threshold for transparency.

1 | INTRODUCTION

Image steganography involves concealing messages within images, as opposed to encryption, which focuses on keeping secret messages unreadable to adversaries. The goal of steganography is to disguise the existence of the message rather than its content [1]. Confidential information is concealed in a manner invisible to the naked eye using this technique. Deep learning technology is recognized as an effective tool for embedding information within images [2]. Image steganography pertains to the practice of concealing confidential information within ordinary objects to generate a stego image, thereby guaranteeing that solely the transmitter and recipient possess knowledge of the secret message [3].

Steganography involves concealing data within a cover media, with the primary goal being to obscure the identity of the sender, recipient, and content of the hidden information, ensuring that only the intended recipient can access and decipher it [4, 5]. The concealment of secret information within cover images through image steganography is executed so seamlessly that concealed data within the stego image remains indiscernible [6]. Numerous techniques have been suggested for concealing confidential information within an image, known as steganographic methods. The widely adopted approach is the least significant-bits (LSB) method, which involves utilizing the least significant bits of the pixels in the cover image to encode the hidden data [7]. Steganography is a method of transmitting data hidden in content through public communication channels without the possibility of an attacker extracting the hidden data [8, 9].

The steganography algorithm's transparency decreases the attacker's likelihood of suspecting the presence of the secret message, making it highly challenging for steganalysis tools to uncover. Reliability means the probability of recovering the message without error despite the transformations and changes done by the steganography algorithm, which must be 100% in steganography methods and therefore is not a decision criterion. Capacity refers to the quantity of data that can be concealed within the steganography content.

Deep learning has become increasingly popular in recent years for its capability to automatically extract features, leading to its widespread use in various modern steganography techniques. In steganography, it is desirable to raise the three factors of transparency, robustness, and capacity as much as possible. Creating methods with transparency and high capacity is particularly important. Our goal is to achieve this by using a neural network.

Steganography has inherently a challenging problem. The algorithm should ensure that the hidden message in the stego image remains undetectable to both human eyes and steganalysis tools, while also being able to accurately decode the message from the stego image without any errors. The task of image steganography has important challenges that we will examine below:

- Necessity of transparency: Transparency can be divided into statistical transparency and visual transparency. Put simply, the message should be undetectable by steganalysis tools or the human eye. Guaranteeing statistical transparency is more difficult than ensuring visual transparency [10]. If there is a noticeable anomaly, the attacker will likely become aware of the secret message's presence. Therefore, it is important to conceal the secret message within the image without introducing any noise or artefacts that could raise suspicion for the attacker.
- Reliability: The algorithm must be able to recover the message fully. Considering that it is probably encrypted before entering the steganography algorithm, the algorithm should not introduce any errors in embedding and retrieving the message [3]. This problem is incompatible with artificial intelligence algorithms because neural networks, like other artificial intelligence methods, usually do not reach 100% accuracy. Therefore, conditions must be provided to correct the errors in the encryption process.
- Challenge for high capacity: It is desirable to increase the number of bits associated with the hidden message in content. Considering that the image contains a lot of information and the intensity of the colour is used for RGB channels in each pixel, the embedding algorithm must keep most of the space to display the image, and as a result, there is not much space to store the secret message [1]. Therefore, achieving high capacity in image embedding is a difficult challenge.
- Incompatible goals: The three factors of transparency, robustness, and capacity are not aligned. Increasing transparency and capacity will reduce robustness, while increasing the robustness of the algorithm will decrease transparency and capacity [11]. Hence, achieving complete transparency, robustness, and capacity simultaneously is unattainable. The steganography algorithm needs to strike a balance among these three elements to ensure they all fall within acceptable limits. Reliability is another concern in neural network solutions. In this research, the Reed–Solomon encoding method [12] is used to fix the recovery error. In this method, it is possible to achieve full reliability by converting the secret message into a larger string. In fact, by converting the secret message into a larger encrypted message and sending it to the steganography algorithm, the message reaches the recipient completely intact. The size of the secret message will increase based on the accuracy of the encoder–decoder network. That is, the more error the encryption algorithm has, the Reed-Solomon algorithm [12] must make a larger string, and as a result, the initial message must be smaller so that the network can hide it. As a result, in the proposed method, reliability is guaranteed at the cost of capacity reduction. In the end, two factors of transparency and capacity remain important, which are in conflict, and the increase of one will cause the decrease of the other.

Our proposed VidaGAN addresses these issues and achieves high capacity with acceptable transparency. With a similar transparency setting, VidaGAN shows a considerable advantage in embedding capacity compared to previous methods.

The following are the contributions of this work:

1. Our deep learning network, built on GAN [13] technology, has achieved state-of-the-art performance.
2. By modifying CSPNet [14], we introduced a novel convolutional backbone to process data.
3. We introduce a new loss function for soft labelling [15] to help prevent the network from overfitting.
4. We introduce a novel method to balance transparency and capacity during training of the neural network.
5. We evaluated the strength of our approach against steganalysis tools to prove our method achieves great transparency with auROC of 0.6.
6. We assessed the practicality of MSE targeting by training our model with a range of MSE targets, showing that adjusting the MSE between cover and stego images can reliably and predictably strike a balance between capacity and transparency.
7. We conducted robustness analysis with JPEG compression, noise, and crop attacks.

This document is presented in four sections. The first section discussed an introduction to the problem, the statement of the problem, goals, challenges, and innovations of the proposed method. Section two will examine existing image steganography techniques to offer a comprehensive look at various approaches to addressing the issue of steganography. In section three the proposed method is presented, which describes how the designed network works and how the training was done. In section four experiments and results including comparison with related methods are presented. Section five delves into conclusions and potential future research directions.

2 | RELATED WORK

This section will cover the existing research and related works in the area of image steganography. The previous approaches can be categorized into traditional methods, methods utilizing convolutional neural networks (CNN), and methods employing GANs.

Traditional steganography: Processing the least significant bit is a widely used method in traditional image steganography. This technique operates under the belief that altering the least significant bits of pixels will not noticeably impact the overall image.

It is important to exercise caution when using the replacement method, as overloading the cover image may result in noticeable changes that could reveal the presence of hidden data [16, 17]. In [18] the secret message is encrypted using Huffman encryption. The bits that have been encoded are then inserted into the least significant bit frame. In addition to the spatial domain, steganography algorithms have also been applied to quantum images [19, 20].

Pixel value difference (PVD) [21] is a traditional technique frequently employed in image steganography. It ensures the consistency of the original image by calculating the variance between adjacent pixels to identify appropriate locations for embedding. The method suggests utilizing a combination of the least significant bits (LSB) in the initial two bits and PVD in the remaining six bits for every 8-bit segment [22]. In [23] the initial local binary patterns (LBP) are combined and rearranged before being compared to generate the steganographic image. In [24], the edges of the cover colour image are obtained rather than using LBP. Medical JPEG images are utilized in this process [25], where local embedding is achieved by comparing the variation in DCT coefficients of matching pixels in adjacent blocks.

Steganography by CNN: CNN models are commonly used in image steganography, utilizing an encoder–decoder structure. The encoder takes the cover image and secret message as input to produce the stego image, which is then fed into the decoder to reveal the hidden message. In [26] and [27] a U-Net [28] based encoder–decoder architecture is used to generate the latent image and a CNN with six layers is used to extract the hidden message. In [29] batch normalization combined with ReLU are used. In this method, SSIM and MSE are applied as loss functions. In [30], ELU and batch normalization are used. In [27], a new cost function, known as variance error, is suggested to minimize the impact of noise in the steganographic image. In [31], the stego image is generated by taking into account the style image, the secret message, and the cover image. In [32], the cover image is transformed into YCbCr format, with the secret message concealed exclusively in the Y channel as the Cr and Cb channels contain all semantic and colour information.

Steganography by GAN: GANs were introduced as a type of deep network in [13]. In the GAN architecture, a generative network and a discriminative network engage in competition to produce a lifelike image. The generative model generates fresh data, while the discriminator network distinguishes between the generated images as either real or fake. In some methods, steganalysis is also considered one of the components of the network. These methods work based on the following three factors: 1) A generator model (G); 2) a discriminator model (D); and 3) a steganalysis model. Three models, G, D, and S, undergo training in an adversarial environment. In [33], introducing steganography using GAN (SGAN), a simplified DCGAN with three modules G, D, and S is presented. A GAN structure consisting of three components is suggested in [34] and [35]. The three models engage in competition where the generator creates the stego image, the discriminator decodes the secret message, and the steganalysis monitors the generator's activity.

3 | PROPOSED METHOD

3.1 | Overview

In this research, we improve image steganography using new and advanced methods. The method being suggested is founded on the GAN framework [13]. The architecture consists of encoder, decoder, and critic [36] networks, where increasing the hiding capacity and improving accuracy, PSNR [37], and SSIM [38] are of particular importance. The encoder and decoder networks utilize the CSPNet [14] architecture as their foundation, which is inspired by the DenseNet network architecture [39] and removes a part of the output of each convolution block from the calculation process, which results in a decrease of computation cost while maintaining the advantages of DenseNet.

The encoder ε is responsible for embedding the secret message M into a stego image S that closely resembles the cover image C. The decoder d's role is to extract the secret message from the stego image. Critic c's role is to verify that the stego image aligns with the distribution of real images P_C. The system, like SteganoGAN [1], works in two ways: At the time of encoding, according to Equation (1), the encoder transforms the cover image and the secret message into the stego image. At the time of decoding, according to Equation (2), The stego image is inputted into the decoder, which then produces the secret message as output M'. An overview of the proposed framework is depicted in Figure 1.

$$S = \varepsilon\left(C, M\right) \tag{1}$$

$$M' = d\left(S\right) \tag{2}$$

Image steganography can be considered as an image processing problem, and modern computer vision methods can be used to solve it. Deep learning methods and especially convolutional networks have reached modern results for many computer vision problems [40–43]. In other words, we can use general-purpose architectures like ResNet [44], CSPNet [14], U-Net [28], etc. for our steganographic goals.

3.2 | Encoder architecture

The encoder creates the stego image using the cover image and secret message, and is tasked with concealing the secret message, which from the GAN's point of view is a generator of fake images. In steganography, it is desirable to increase the hiding capacity, but the hidden image should not be distorted in such a way that it can be identified by a human attacker or a steganalysis tool [3]. In other words, increasing the capacity should not cause a sharp decrease in transparency. Considering the difficulty of this goal, the encoder must have high learning capability to discover patterns and use many local relationships between pixels.

In image steganography, like other computer vision problems, unnecessary processing should be avoided. Data should enter a processing layer only if it is deemed necessary. ResNet

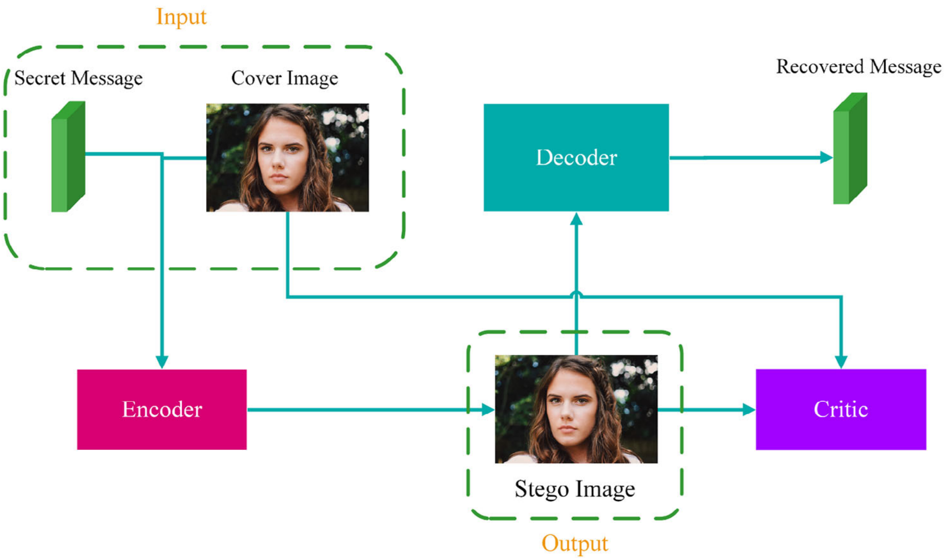


FIGURE 1 The overall architecture of VidaGAN. The system converts the cover image and the secret message into a stego image, which is sent through public communication channels and later used by the decoder to recover the original secret message.

[44] was able to meet this requirement to some extent by adding a shortcut between the beginning and end of each block. DenseNet [39] took a more aggressive approach by incorporating shortcuts from the output of each convolution layer to the input of all following layers. In this research, the design of the encoder architecture is inspired by CSPNet [14], which removes a part of the output of each block from the rest of the calculation process. This reduces the complexity of calculations while maintaining the accuracy of DenseNet.

In VidaGAN the encoder architecture is a modified CSPNet. The idea of CSP is to leave part of the processed information from the rest of the computation. Thus, after each convolution block, part of its output channels goes directly to the last convolutional layer. As a result, the gradients for this data will not pass through the additional layers during the backward propagation [14]. Due to the finalization of part of the information after each block, less calculation is performed in the subsequent block, and repetitive and useless operations are avoided. As a result, the computational complexity and memory requirement will be reduced.

However, there are differences between the VidaGAN encoder and CSPNet. In CSPNet dense blocks are used but the VidaGAN encoder utilizes simple convolutional blocks. Another difference is that the VidaGAN encoder does not reduce resolution whereas CSPNet does it after every stage. Figure 2 shows a convolutional encoder block. The output of each convolutional layer has 48 channels, which is more than CSPNet. Additionally, in the VidaGAN encoder, removing half of the block's output channels from further processing automatically results in block specialization, eliminating the need for a transition layer to specialize blocks.

Figure 3 shows the encoder architecture. The secret message has six channels because, for every pixel, six bits are stored. Each trapezoid represents a convolution block. The output of each block has 48 channels, 24 of which will participate in the continuation of the network (blue) and 24 will be separated from the rest of the process (magenta). Blue layers will be processed in all subsequent blocks, which is similar to DenseNet [39] behaviour, and magenta layers will not participate in subsequent convolutions, which is CSP behaviour. In the last convolution, the cover image, the secret message, and all the blue and magenta outputs are concatenated and sent to the last convolution with 201 input channels and three output channels. These three channels form the stego image.

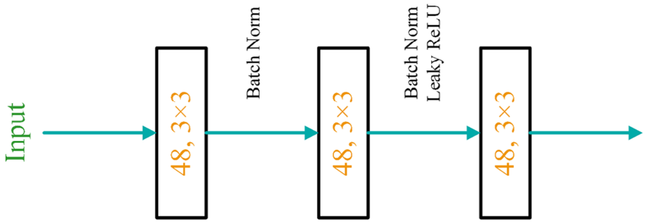


FIGURE 2 Convolutional block in used VidaGAN encoder and decoder.

3.3 | Decoder architecture

Decoder architecture is similar to the encoder. The only difference between encoder and decoder architecture is in input and output. The architecture of the decoder is illustrated in Figure 4, which is responsible for extracting the secret message from the stego image input.

3.4 | Critic architecture

The critic's architecture is depicted in Figure 5. In each iteration, a batch of cover images is initially presented to the

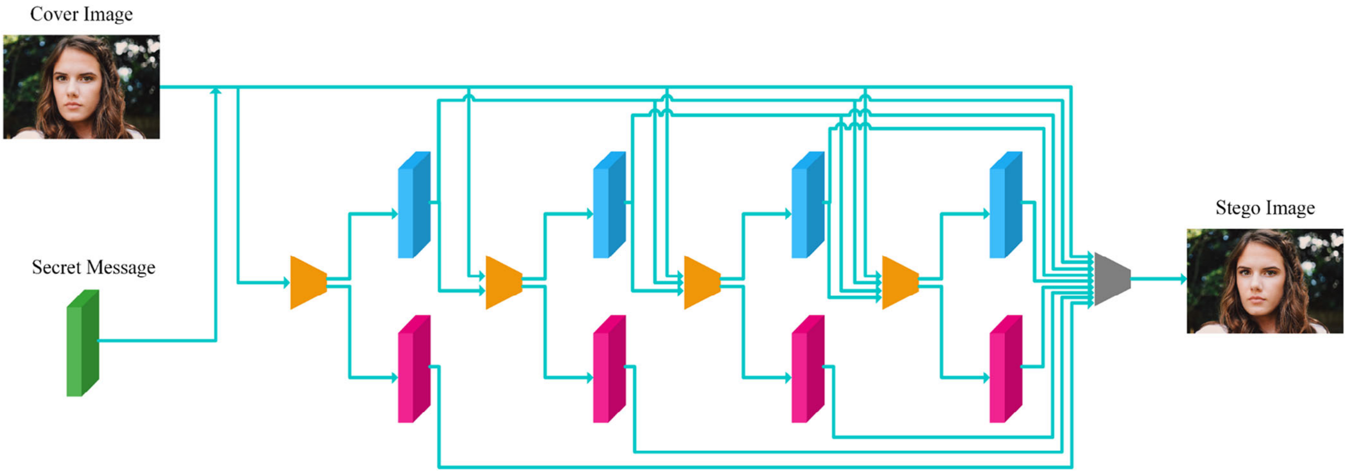


FIGURE 3 The architecture of the encoder.

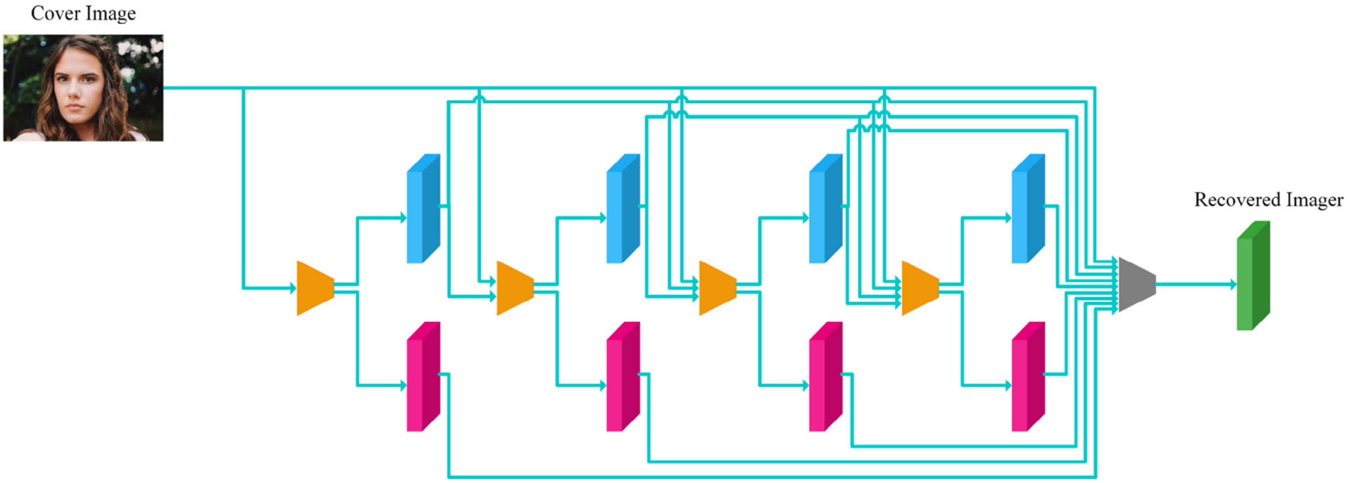


FIGURE 4 The architecture of the decoder.

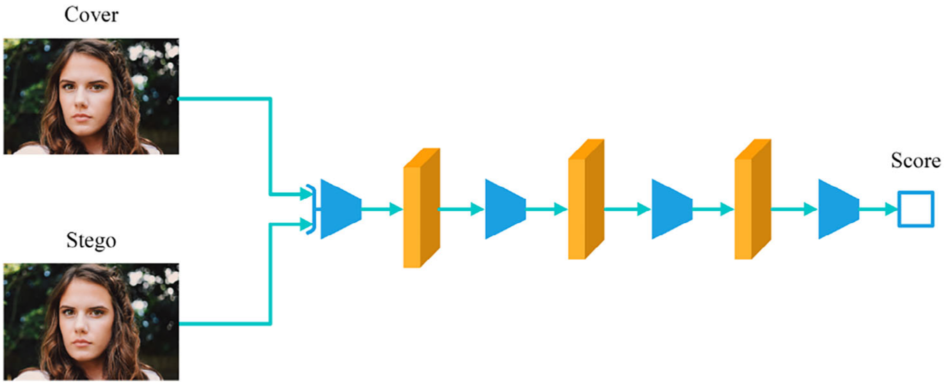


FIGURE 5 The architecture of the critic.

critic to assess their authenticity. Then, a batch of stego images is given to detect fakes. The first three convolutions have 32 output channels, and the last convolution has one output channel. The kernel size of all convolutions is 3 × 3. After outputting the result, its average is calculated as a scalar number. This number is the critic score that can be used to train the network with the Wasserstein loss function [36].

During the design of the critic architecture, the following points were taken into consideration:

1. To recognize whether the image is real or fake, one should pay attention to the existence of point noises similar to salt-and-pepper noise. A critic which carefully observes local characteristics can easily detect the presence of salt-and-pepper noise, prompting the encoder to take measures to prevent noise from being created.
2. Due to the point-based nature of the salt-and-pepper noise, reducing the processing resolution will remove some of these noises. The critic's inability to observe these noises will disrupt the training process. As a result, the processing occurs at the resolution of the input image.
3. Due to the locality of the features, there is no need to extract complex and high-level features and as a result, processing can be done in fewer steps.

3.5 | Network loss

In order to minimize the difference between the output image generated by the encoder (stego) and the original input image (cover), an MSE loss function is employed for the generator, as outlined in Equation (3). For adversarial training, according to Equation (4), Wasserstein [36] loss is applied. According to Equation (5), a binary cross-entropy loss is used for the decoder, which aims to ensure the correctness of the recovered message.

$$L_{\varepsilon} = \frac{1}{3 \times W \times H} \sum \left(C - \varepsilon\left(C, M\right)\right)^2 \tag{3}$$

$$L_{adv} = \begin{cases} \textit{for generator}: -c\left(\varepsilon\left(C, M\right)\right) \\ \textit{for critic}: c\left(\varepsilon\left(C, M\right)\right) - c\left(C\right) \end{cases} \tag{4}$$

$$L_d = \textit{crossentropy}\left(d\left(\varepsilon\left(C, M\right)\right), M\right) \tag{5}$$

3.6 | Soft labelling

In the cross-entropy function, putting values of zero and one as labels can disrupt the training of the network, because it causes the network to be overconfident in its estimates. Vanishing gradients occur when the predicted output is in close proximity to the actual label. This widens the range of inputs for the sigmoid activation function, but due to the senselessness of this wide range, further optimization will only cause the network to overfit.

Using soft labels can be considered a regularization method like L1 and L2 [45] to avoid overfitting. In this research, a new experimental loss function is introduced according to the relations (6) and (7), which work as a soft label method. This loss function is more flexible than popular soft label methods such as label smoothing [15] due to having two hyperparameters.

$$L_{soft} = \propto X^{\gamma} \tag{6}$$

$$X = 2\sigma - 1 \tag{7}$$

where σ is the output value after applying the sigmoid, $\propto$ and γ are hyperparameters, and X is the sigmoid output converted to the range $[-1, 1]$.

As evident in Figure 6, the value of the soft label function is minimal in the neighbourhood of zero. As the distance from zero increases, the value of the function increases. By setting hyperparameters as $\propto = 0.15$ and $\gamma = 4$, We achieved the best result.

3.7 | MSE targeting

The generator loss is calculated by adding the loss functions of various components of the network as specified in Equation (8):

$$L_{gen} = L_{\varepsilon} + \theta\left(L_{adv-gen} + L_d + L_{soft}\right) \tag{8}$$

Considering the adversarial relationship of the encoder with the critic and the decoder with a hyperparameter θ, the relative strength between these two fronts is determined. The goal of the encoder loss L_{ε} is to minimize the difference between the pixels of the stego image and the cover image. Giving excessive importance to this loss will lead to a significant penalty for any variation in pixel intensity between the original and the hidden images. As a result, the network will be forced to limit the changes to the pixel values in a very small range. This, in turn, will reduce the hiding capacity of the network and the accuracy of recovery. On the contrary, a low value will greatly increase the accuracy of recovery, but the stego image will appear distorted and different from the original cover image due to the expanded range of pixel changes. Even with the implementation of GAN architecture, encoder loss is still necessary because of the need for similarity between cover and stego images.

The hyperparameter θ specifies the strength of the encoder compared to the decoder and the critic. This parameter can be considered fixed like SteganoGAN [1] or it can be considered variable and adaptive. In VidaGAN, a new technique called MSE targeting is introduced, which aims to keep the stego image quality constant by setting a target value for the encoder loss L_{ε}. With this approach, the stego image quality and as a result transparency is determined.

The value of θ varies during training. By adjusting it, the encoder loss value can be kept at a somewhat constant level. In this section, the encoder loss function L_{ε} is named MSE for simplicity. The current value of θ is determined based on previous iterations. The main idea of the MSE targeting algorithm is based on the linear fit of $log(MSE)$ for previous iterations, and the value of θ is determined according to the slope of the fitted line.

In Figure 7, the red series shows the value of $log(MSE)$ in previous iterations. In the example in this figure, the current iteration is 360. First, according to the previous 60 iterations (from 300 to 360), the blue line is fitted. This line shows the current trajectory of MSE with the current value of θ. It is assumed that

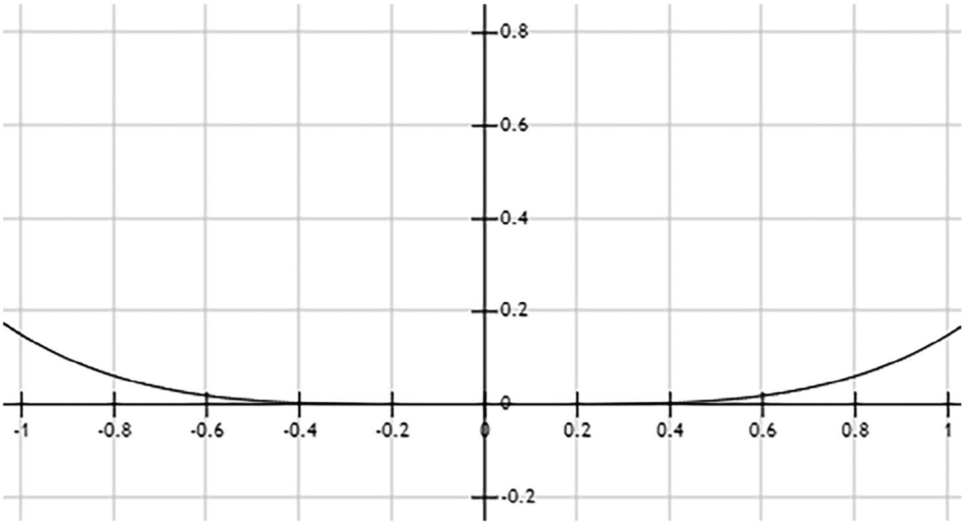


FIGURE 6 The proposed soft label function.

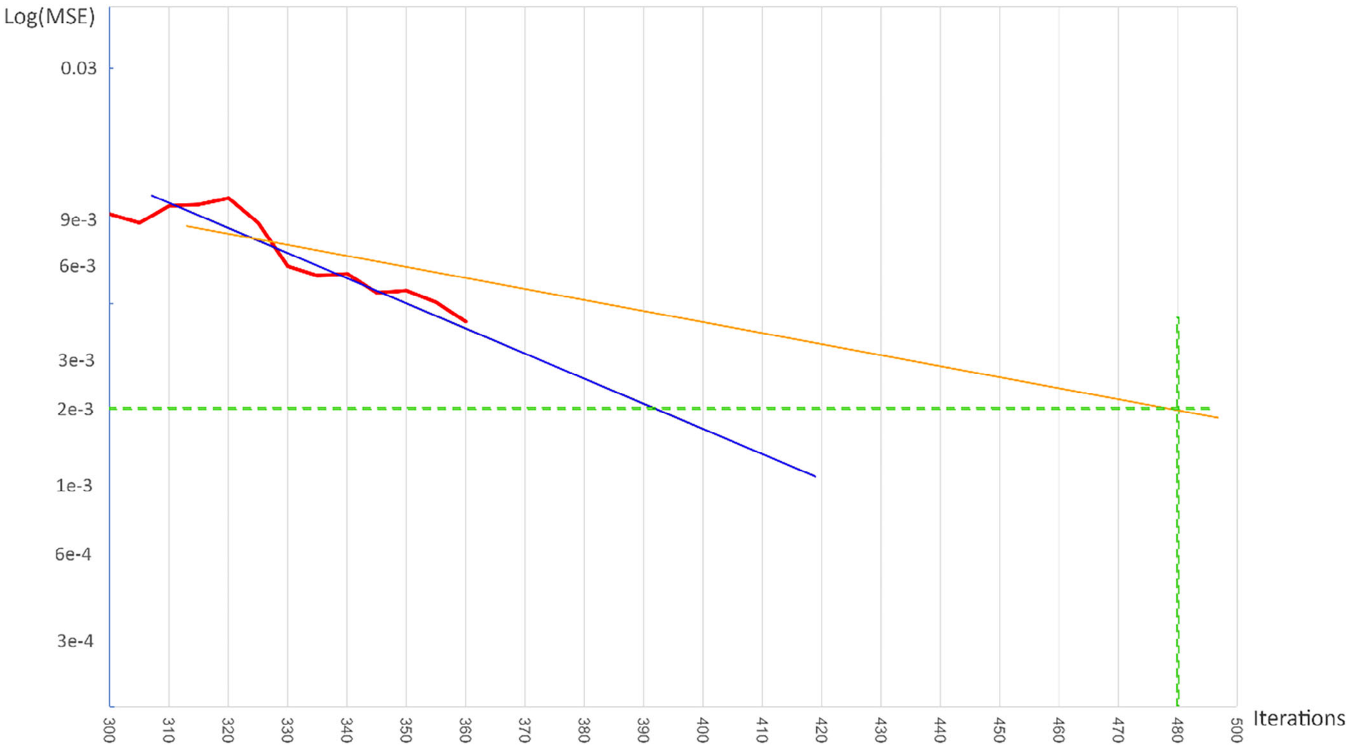


FIGURE 7 MSE targeting.

reaching the target MSE value after 120 iterations is desirable. In this example, the target MSE value is 0.002. The horizontal green dotted line shows the target MSE, and the vertical green dotted line marks 120 iterations later. The orange line is drawn from the horizontal and vertical average of the last 60 iterations to the intersection of the two green dotted lines. The slope of this line guides the algorithm to reach the target MSE in the next 120 iterations. The ideal situation is the complete matching of the blue and orange lines. If the blue line is lower than the orange line on the right side of Figure 7, it means MSE is decreasing at a faster rate than required. In this case, θ should be increased so that the blue line approaches the orange line in the future. Conversely, if the blue line is above the orange, θ should decrease. By doing this adjustment in each iteration, the MSE value gradually falls into the target range. The method of implementing this technique is based on considering the difference in the slope of two lines. The value of θ is adjusted according to the relations (9) and (10).

$$\theta \leftarrow \theta\,(1 + \Delta m) \tag{9}$$

$$\Delta m = m_{orange} - m_{blue} \tag{10}$$

According to the mentioned method, MSE can be directed towards the target. In addition to adjusting the MSE value, this technique for targeting MSE in some settings slightly increases the decoder accuracy. The reason for this is the disturbance of the balance between the encoder and decoder and the introduction of noise to the system by varying θ. Introducing noise to the network in some cases such as [42] and [46] may increase the accuracy. By making changes in θ, the network sometimes makes the encoder stronger and sometimes the decoder.

3.8 | Training

VidaGAN network is trained for 16 epochs. The Adam optimizer [47] is selected with a learning rate of 0.0002. Training starts with a warm-up epoch. From the second epoch onwards, the learning rate decreases linearly until it reaches 1.5×10^4 in the last epoch. Horizontal flips and random crop were used as data augmentation. For each iteration, as in step one, one batch of cover images and another batch of stego images are sent to the critic, and Equation (4) is used as the loss function. In this step, only critic parameters are updated. For the second step, the encoder and decoder are trained with the loss function defined by Equation (8). In the training process, a tensor of Bernoulli random numbers is used to simulate the secret message.

3.9 | Recovery error correction

The encoder–decoder network has state-of-the-art accuracy in recovering the message. However, the recovery accuracy will not be 100%, and some bits will be recovered incorrectly in the decoder. In steganography, erroneous recovery of data is unacceptable. To solve this problem, the Reed-Solomon error correction method [12] is used, which, upon receiving a bit array, returns a larger bit array containing error correction information. Using this method will fix the error in the recovered message. In other words, if a percentage of bits are wrong, the Reed-Solomon algorithm restores the message correctly due to the insertion of error correction information. The increased size of the message is determined according to Equation (11). If the condition holds, the message can be recovered.

$$p.n \leq \frac{n - k}{2} \tag{11}$$

where p is the probability of error in each bit, k is the length of the input bit array, and n is the length of the array augmented with error correction information. The probability of each bit being correct is equal to the recovery accuracy of the decoder. Therefore, by knowing p and k, the minimum value of n can be determined to guarantee the complete recovery of the secret message. First, the bit array of the message is sent to the Reed-Solomon algorithm, and the augmented bit array is generated. This augmented bit array will be sent as the secret message to the neural network. At last, the array received from the decoder is sent to the Reed-Solomon algorithm and the secret message is recovered.

3.10 | Evaluation metrics

To evaluate the quality of the stego image, a comparative quantitative measure can be used between the cover and stego images. The MSE loss function can also be used as an evaluation metric. The PSNR metric [37] considers the intensity of pixel noises as a quality criterion. The SSIM metric [38] gives importance to the structural similarity of two images. All three criteria presented in Equations (12–14) are suitable for evaluating the proposed method and are reported in the results section of this research.

$$\mathrm{MSE} = \frac{1}{W \times H}\,(y - \hat{y})^2 \tag{12}$$

$$\mathrm{PSNR} = 10 \times \log_{10} \frac{s^2}{\mathrm{MSE}} \tag{13}$$

where s is the maximum possible value for the difference of a pixel value in two images.

$$\mathrm{SSIM} = \frac{(2\mu_X\mu_Y + C_1)\,(2\sigma_{XY} + C_2)}{\left(\mu_X^2 + \mu_Y^2 + C_1\right)\left(\sigma_X^2 + \sigma_Y^2 + C_2\right)} \tag{14}$$

where $C_1 = 0.0001$ and $C_2 = 0.0009$ are small numbers for numerical stability. The output values are in the range [−1,1].

The bit-per-pixel capacity is calculated in accordance with the bit depth of the secret message, the recovery accuracy of the network, and Equation (11). To establish this equation, the value of k must be less than or equal to $(1 - 2p)n$. Assuming a 6-bit depth for the secret message, the capacity of the encryption algorithm will be equal to $6(1 - 2p)$. This amount is called Reed-Solomon bit-per-pixel (RS-BPP) and is used for evaluation of results.

4 | EXPERIMENTS

Python programming language and Pytorch framework have been used to implement the proposed method. The existing implementation of the SteganoGAN [1] has been used for comparison. A NVidia RTX 3060 GPU has been used for training and evaluation of the network. For the data, we chose div2k [48], which has 800 images for training, 100 images for validation, and 100 images for testing. The depth of the secret message tensor has been tested for 1 through 6 bits. The 6-bit mode gives the best answer.

4.1 | Main results

The steganography results are shown in Table 1. In this table, the recovery accuracy, MSE, PSNR, and SSIM are

TABLE 1 VidaGAN output for different bit depths.

Bit depth	Recovery accuracy	Capacity (RS-BPP)	PSNR	SSIM
1	**1.0**	0.99	**39.07**	**0.930**
2	0.996	1.98	36.92	0.900
3	0.977	2.86	37.31	**0.930**
4	0.927	3.42	36.15	0.908
5	0.868	3.68	36.08	0.899
6	0.825	**3.90**	38.56	0.884

Abbreviations: PSNR, peak signal noise ratio; RS-BPP, reed-solomon bit per pixel; SSIM, structural similarity index measure.

TABLE 2 Comparison between VidaGAN and other methods in the div2k dataset with a payload of four bits per pixel.

Method	Recovery accuracy	PSNR	SSIM
SteganoGAN [1]	0.82	37.49	0.88
Coverless [51]	0.789	35.35	0.85
FNNS-R [49]	0.891	28.60	0.76
FNNS-D [49]	0.945	25.74	0.65
Secure FNNS [50]	0.912	25.79	0.77
Ours	**0.970**	**37.51**	**0.90**

Abbreviations: PSNR, peak signal noise ratio; SSIM, structural similarity index measure.

reported between cover and stego images. Also, the results of SteganoGAN [1] are presented for comparison. With a depth of 6 bits for the secret message, the highest capacity has been achieved as the image can store 3.9 bits in each pixel of the image. This result has improved by 60% compared to the value of 2.44 by SteganoGAN for div2k data.

In Table 1, bit per pixel has an upward trend when increasing the bit depth. Based on these results, the bit depth of 6 is the best answer for VidaGAN. It should be noted that with increasing bit depth, the accuracy of recovery decreases. This is due to the need to hide more data. In other words, recovery accuracy and capacity, as expected, are non-aligned goals, and increasing one will decrease the other.

Looking into the columns of MSE, PSNR, and SSIM, it is clear that their value does not change much with the change of bit depth. The reason is MSE targeting technique stabilizes the range on MSE. Keeping the stego image quality somewhat constant eliminates one of the key variables (transparency) and makes the results easier to interpret.

In Table 2, the proposed method is compared with previous methods on div2k dataset for payload with 4 bits per pixel. The proposed method offers better accuracy and quality. FNNS [49] and Secure FNNS [50] have less accuracy than the proposed method while horribly degrading the image in terms of PSNR and SSIM. We contend that our results are superior, achieving a balance between capacity and quality. Our best 4-BPP result was achieved with a target MSE of 0.001. Section 4.2 demonstrates that our method allows for precise control over the trade-off between capacity and quality when experimenting with MSE targeting.

TABLE 3 Ablation studies for each contribution (6-BPP).

Network	Recovery accuracy	RS-BPP	PSNR	SSIM
SteganoGAN [1]	0.70	2.44	38.94	**0.9**
CSP Architecture	0.767	3.20	**38.96**	**0.90**
+ Soft Labeling	0.817	3.81	37.45	0.89
+ MSE targeting	**0.825**	**3.90**	38.56	0.88

Abbreviations: PSNR, peak signal noise ratio; RS-BPP, reed-solomon bit per pixel; SSIM, structural similarity index measure.

4.2 | MSE adjustment

MSE targeting tries to control the quality of the stego image by adjusting the discrepancies between cover and stego images, as presented in Section 3.7. To evaluate its effectiveness, we have performed several runs with different target MSE and 4-BPP payload. Figure 8a–c shows accuracy, PSNR, and SSIM for each target MSE. The negative correlation of accuracy with PSNR and SSIM is obvious and smooth. It also shows the effectiveness of MSE targeting as a reliable tool to balance between capacity and quality.

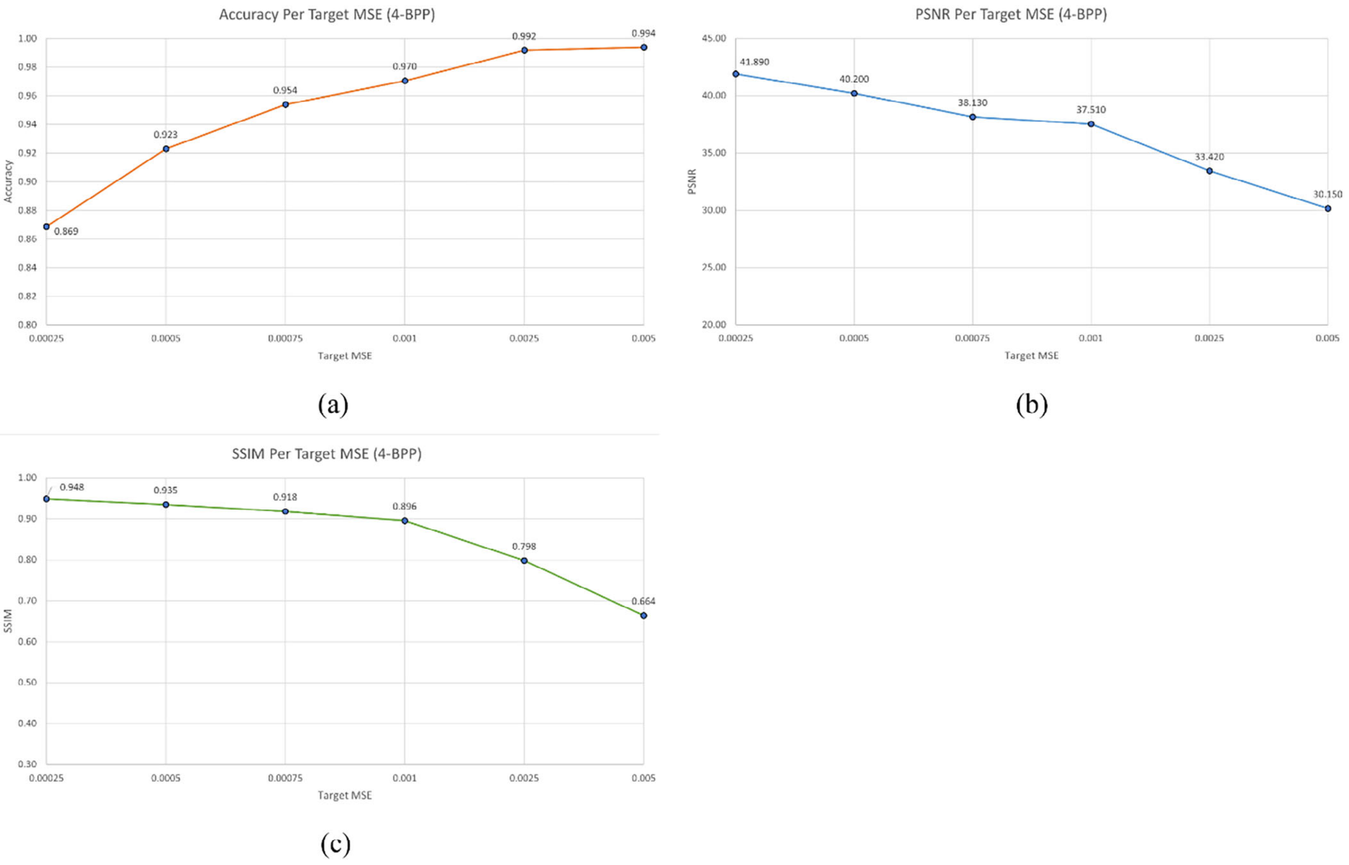
With Reed-Solomon encoding (Section 3.9), capacity is directly derived from accuracy. Figure 9a,b illustrates the relationship between capacity and both MSE and PSNR. As shown, higher MSE results in lower image quality but increased capacity, whereas higher PSNR indicates better image quality but reduced capacity.

4.3 | Ablation studies

VidaGAN is competitive with other steganography methods. The three main contributions of 1) CSP [14] architecture design, 2) soft labelling, and 3) MSE targeting in this research have increased the accuracy of neural network recovery and thus the network's hiding capacity. In this section, the effectiveness of each is checked according to Table 3. The architecture of SteganoGAN [1] is taken as the baseline and the contributions are applied sequentially. First, the CSP architecture is applied, and then the soft label is added to it. Finally, MSE targeting is applied.

SteganoGAN uses a dense block with three layers in the encoder and decoder. In this research, by designing an architecture based on CSPNet [14], high network complexity, and capacity are obtained. The resulting network, being superior to the SteganoGAN network, brings the capacity to 3.20 bits per pixel.

Next, the proposed soft labelling is applied to the new architecture. Due to the binary nature of the decoder results, it can be solved as a binary classification problem. Soft labelling can be used in any classification problem. The soft label as a network

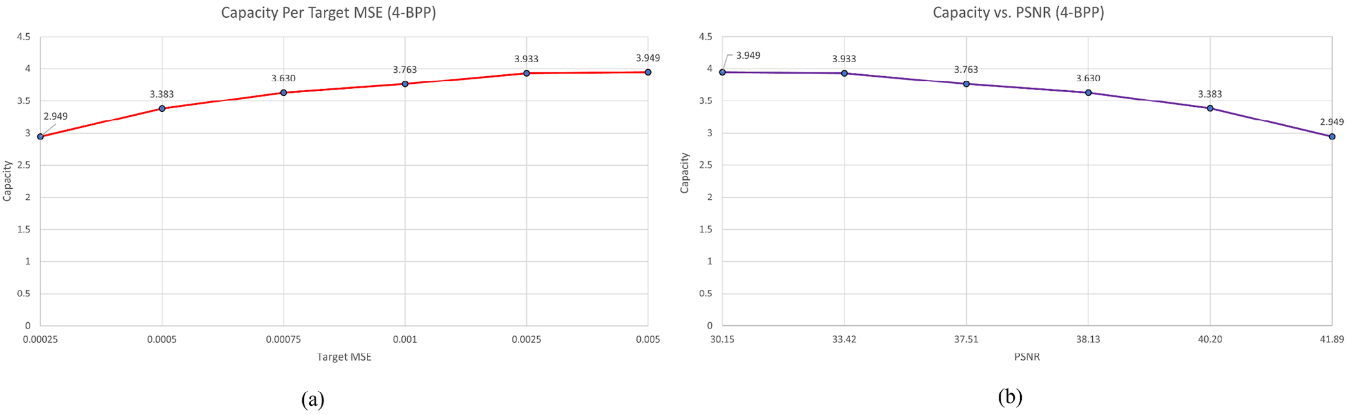


(a)

(b)

(c)

FIGURE 8 The effect of setting target MSE on: (a) Accuracy, (b) PSNR, (c) SSIM.



(a)

(b)

FIGURE 9 Capacity vs (a) MSE, (b) PSNR.

normalizer may increase the accuracy of the method by preventing overfitting. In this case, the bit-per-pixel value is raised to 3.81.

Then we add MSE targeting. This technique adjusts the output quality by keeping the MSE error approximately constant between the cover and the stego images. In addition, due to the noise of the network, the accuracy is increased and the hiding capacity reaches 3.90 bits per pixel.

4.4 | Throughput analysis

We conducted a throughput test to evaluate the time required for embedding a secret message into a cover image and for extracting the recovered message from a stego image. The embedding process is very quick, capable of processing multiple images per second. While the decoder network operates swiftly during extraction, the message recovery process

TABLE 4 Computation time for images with 4-BPP.

	Time taken (s)
Embedding	0.2
Extraction	1.08
Embedding + Extraction	1.28

TABLE 5 Time to embed and extract data with 4-BPP.

	Time taken (s)
SteganoGAN	0.09
FNNS-R	159.19
FNNS-D	44.29
LISO	0.33

performed by Reed-Solomon [12] decoding is comparatively slow. Table 4 presents the time it takes to embed and extract images.

Table 5 details the efficiency of previous methods run on Titan RTX GPU [52] showing that our method is slower than SteganoGAN and LISO [52] while considerably faster than FNNS [49].

4.5 | Steganalysis

The purpose of steganography is to hide the message from detection. Steganalysis is responsible for discovering the hidden message in the content under investigation. One of the popular steganalysis tools is StegExpose [53]. This tool determines the existence of a hidden message in an image using methods such as RS analysis [54], sample pairs [55], and primitive sets [56] according to the least significant bit.

RS analysis detects hidden data in black and white and colour images by examining the difference between the number of single and collective groups for the least significant bit and shifted least significant bit planes. Analysis of sample pairs is based on finite-state machines. The primitive sets work based on identifying a statistical identity related to a series of image pixels.

In this research, 100 masked and hidden images were given to StegExpose [53]. StegExpose tries to detect images with hidden messages by using the mentioned algorithms. Figure 10 shows the ROC diagram for hidden message detection with an AUC of 0.6. As a result, StegExpose prediction is not much better than random guessing. In SteganoGAN [1], this value is equal to 0.59.

The reason for achieving the desired transparency is the use of the encoder and Critic loss functions, which leads to limiting the range of changes in the stego image and the realism of the output. Therefore, considering the smallness of the changes between the cover and stego images, distinguishing between them will not be an easy task.

4.6 | Robustness

Robustness refers to a steganography method's resistance to attacks. During the transmission of secret data or payloads, the algorithm can fail due to various attacks, including noise, scaling, and JPEG compression. The robustness of the proposed method was assessed, evaluated, and confirmed through experiments and comparisons [57]. We evaluate the robustness of our method with bit error rate (BER) defined by Equation (15) [57],

$$\text{BER} = e/n, e = \sum_{i}^{n} p_i \neq q_i \tag{15}$$

where e denotes the number of errors detected, n represents the total number of bits, p is the vector of secret message recovered without the attack, and q is the vector of secret messages recover with the attack. If BER is 0, it means no errors were found, and the secret bits were extracted with 100% accuracy, demonstrating that the method is completely robust against this attack. However, if BER > 0, there is an error rate in the extracted secret bits post-attack (i.e., some secret bits have been altered or damaged), indicating that the method is not fully robust against this attack.

We conducted noise, crop, and JPEG compression attacks [58] to evaluate the robustness of the proposed method. In the crop attack, we inserted zeros into the margins of the stego image, resulting in complete data loss. The width of the zero margins was varied at 5, 10, and 15 pixels. For the noise attack, random salt and pepper noise were introduced into the stego image, with noise densities of 0.01, 0.02, and 0.03. The JPEG compression attack involved applying compression to the stego image with qualities of 90, 70, and 50. The results of these experiments are presented in Table 6.

As indicated by the numerical results, it is clear that the proposed method exhibits vulnerability to attacks and modifications. This outcome is consistent with our objective of having a high-capacity and transparent steganography algorithm. While this may be perceived as a limitation, it's worth noting that most steganography studies do not emphasis on robustness [1, 50, 59] as improving robustness negatively impacts capacity. This stands in contrast to image watermarking, where the importance of a robust algorithm is paramount [60, 61].

5 | CONCLUSION AND FUTURE WORK

This study introduces a novel method called Variable Adaptive GAN (VidaGAN) for embedding arbitrary binary data into images using generative adversarial networks. VidaGAN enhances the visual quality of the resulting stego images, achieving high PSNR and SSIM. Besides improving the perceptual quality of the images generated by our model, we also attained higher accuracy in the results. Our 4-BPP setup surpasses previous methods in both accuracy and quality, while our 6-BPP setup achieves a state-of-the-art capacity of 3.9 bits per pixel on div2k, producing high-quality results that are resistant to steganalysis.

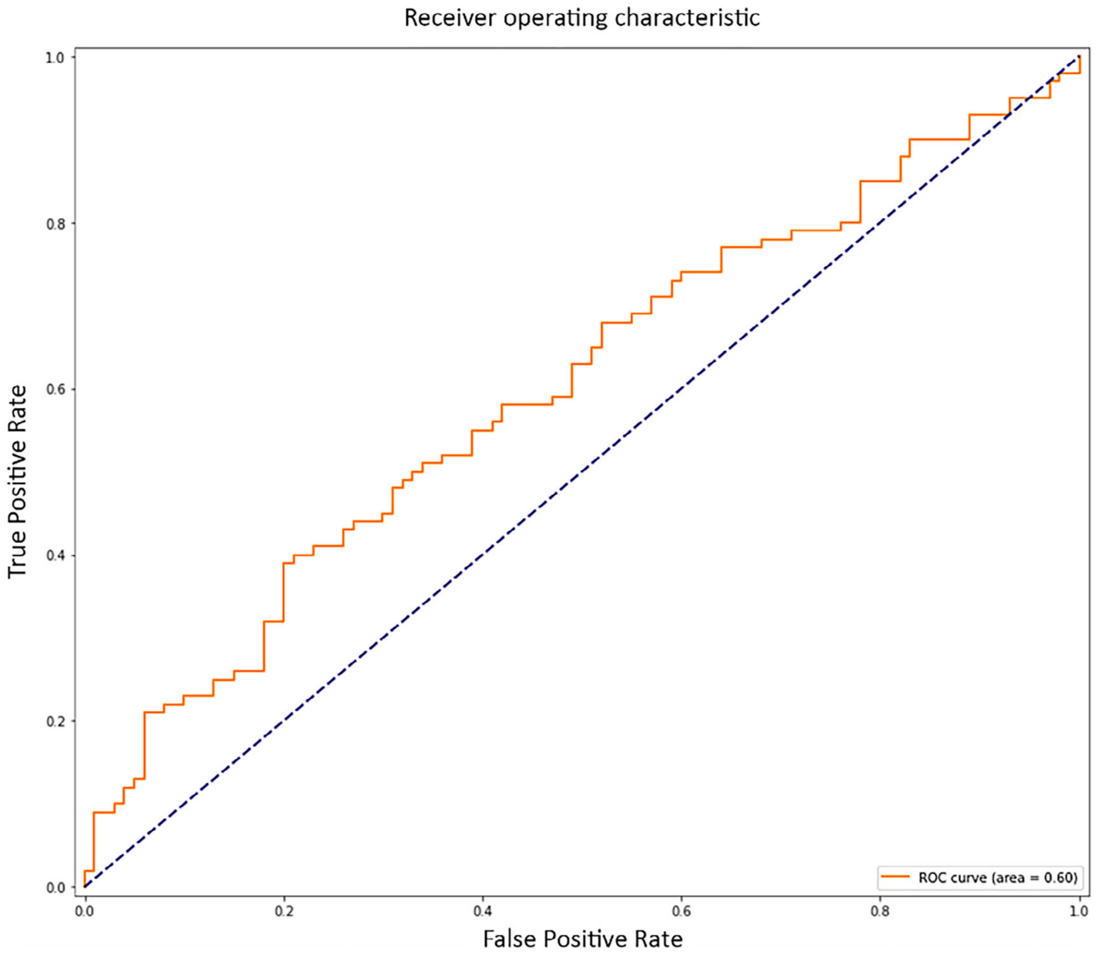


FIGURE 10 ROC curve for hidden message detection.

TABLE 6 Bit error rate caused by different attacks.

	Noise (%)			Crop (pixels)			JPEG compression (quality)		
	0.01	**0.02**	**0.03**	**5**	**10**	**15**	**90**	**70**	**50**
BER	0.214	0.304	0.354	0.050	0.076	0.100	0.456	0.484	0.490

However, this work can be further extended to alleviate some of its limitations. The robustness of the proposed method can be improved against JPEG compression, crop, and noise attacks by including them in the training process. The extraction phase is not as efficient. Visual transformers can be utilized to enlarge the receptive field and augment the parameter count for the neural network. Another direction of improvement is to adopt CycleGAN instead of basic GAN architecture to take advantage of reconstruction loss as a mechanism to enforce transparency. With this approach, it would be possible to balance the reconstruction and adversarial losses with the help of MSE targeting introduced in this work.

AUTHOR CONTRIBUTIONS

Vida Yousefi Ramandi: Data curation; formal analysis; software; visualization; writing—original draft. **Mansoor Fateh**: Conceptualization; methodology; supervision; visualization; writing—original draft; writing—review and editing. **Mohsen Rezvani**: Investigation; project administration; resources; validation; visualization; writing—review and editing.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data sharing not applicable no new data generated Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Mansoor Fateh https://orcid.org/0000-0003-2133-3480

REFERENCES

1. Zhang, K.A., et al.: SteganoGAN: High capacity image steganography with GANs. arXiv:1901.03892, (2019)
2. Subramanian, N., et al.: Image steganography: A review of the recent advances. IEEE Access 9, 23409–23423 (2021)
3. Fateh, M., Rezvani, M.: An email-based high capacity text steganography using repeating characters. Int. J. Comput. Appl. 43(3), 226–232 (2021)
4. Rai, P., Gurung, S., Ghose, M.: Analysis of image steganography techniques: A survey. Int. J. Comput. Appl. 114(1), 11–17 (2015)
5. Khan, M., et al.: Image steganography using uncorrelated color space and its application for security of visual contents in online social networks. Future Gener. Comput. Syst. 86, 951–960 (2018)
6. Kim, C., et al.: Blind decoding of image steganography using entropy model. Electron. Lett. 54(10), 626–628 (2018)
7. Wu, H.-C., et al.: Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proc.-Vision, Image Signal Process. 152(5), 611–615 (2005)
8. Provos, H., Honeyman, P.: Hide and seek: An introduction to steganography. IEEE Secur. Privacy 1, 32–44 (2003)
9. Shafi, I., et al.: An adaptive hybrid fuzzy-wavelet approach for image steganography using bit reduction and pixel adjustment. Soft Comput. 22, 1555–1567 (2018)
10. Ballesteros, L.D.M., Moreno A, J.M.: Highly transparent steganography model of speech signals using efficient wavelet masking. Expert Syst. Appl. 39(10), 9141–9149 (2012)
11. Fateh, M., Mohsen, R., Yasser, I.: A new method of coding for steganography based on LSB matching revisited. Secur. Commun. Netw. 2021, 1–15 (2021)
12. Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. J. Soc. Ind. Appl. Math. 8(2), 300–304 (1960)
13. Goodfellow, I., et al.: Generative adversarial nets. Adv. Neural Inf. Process. Syst. 2, 2672–2680 (2014)
14. Wang, C.-Y., et al.: CSPNet: A new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 390–391. IEEE, Piscataway, NJ (2020)
15. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 4694–4703. Curran Associates Inc., Red Hook, NY (2019)
16. Johnson, F.N., Jajodia, S.: Exploring steganography: Seeing the unseen. Computer 31, 26–34 (1998)
17. Gupta, S., Gujral, G., Aggarwal, N.: Enhanced least significant bit algorithm for image steganography. Int. J. Comput. Eng. Manage. 15, 40–42 (2012)
18. Das, R., Tuithung, T.: A novel steganography method for image based on Huffman encoding. In: Proceedings of the 2012 3rd National Conference on Emerging Trends and Applications in Computer Science, pp. 14–18. IEEE, Shillong, India, (2012)
19. Qu, Z., et al.: A novel quantum image steganography algorithm based on exploiting modification direction. Multimedia Tools Appl. 78, 7981–8001 (2019)
20. Wang, S., et al.: Least significant qubit (LSQb) information hiding algorithm for quantum image. Measurement 73, 352–359 (2015)
21. Hong, W., Chen, T.-S.: A novel data embedding method using adaptive pixel pair matching. IEEE Trans. Inf. Forensics Secur. 7(1), 176–184 (2011)
22. Swain, G.: Very high capacity image steganography technique using quotient value differencing and LSB substitution. Arabian J. Sci. Eng. 44, 2995–3004 (2019)
23. Qiu, A., et al.: Coverless image steganography method based on feature selection. J. Inf. Hiding Privacy Prot. 1, 49 (2019)
24. Rashid, R.D., Majeed, T.F.: Edge based image steganography: Problems and solution. Proceeding of 2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 1–5. IEEE, (2019)
25. Liao, X., et al.: Medical JPEG image steganography based on preserving inter-block dependencies. Comput. Electr. Eng. 67, 320–329 (2018)
26. Wu, P., Yang, Y., Li, X.: Image-into-image steganography using deep convolutional network. In: Proceedings of the 2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 792–802. IEEE, Piscataway, NJ (2018)
27. Wu, P., Yang, Y., Li, X.: StegNet: Mega image steganography capacity with deep convolutional network. Future Internet 10, 54 (2018)
28. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical image computing and computer-assisted intervention, pp. 234–241. Springer, Cham (2015)
29. Van, T.P., Dinh, T.H., Thanh, T.M.: Simultaneous convolutional neural network for highly efficient image steganography. In: Proceedings of the 2019 19Th international symposium on communications and information technologies (ISCIT), pp. 410–415. IEEE, Piscataway, NJ (2019)
30. Isola, P., et al.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134. IEEE, Piscataway, NJ (2017)
31. Rahim, R., Nadeem, S.: End-to-end trained CNN encoder-decoder networks for image steganography. In: Proceedings of the European Conference on Computer Vision, pp. 723–729. Springer, Cham (2018)
32. Baluja, S.: Hiding images in plain sight: Deep steganography. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 2069–2079. Curran Associates Inc., Red Hook, NY (2017)
33. Volkhonskiy, D., Borisenko, B., Burnaev, E.: Generative adversarial networks for image steganography. openreview.net (2016)
34. Shi, H., et al.: Synchronized detection and recovery of steganographic messages with adversarial learning. In: Proceedings of the International Conference on Computational Science, pp. 31–43. Springer, Cham, Switzerland (2019)
35. Im, D.J., et al.: Generating images with recurrent adversarial networks. arXiv:1602.05110, (2016)
36. Arjovsky, M., Soumith, C., Bottou, L.: Wasserstein generative adversarial networks. Proc. Int. Conf. Mach. Learn. 70, 214–223 (2017)
37. Zhou, W., et al.: Image quality assessment: From error visibility to. IEEE Trans. Image Process. 13(2), 600–612 (2004)
38. Setiadi, D.R.I.M.: PSNR vs SSIM: Imperceptibility quality assessment. Multimedia Tools Appl. 80(6), 8423–8444 (2021)
39. Huang, G., et al.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700–4708. IEEE, Piscataway, NJ (2017)
40. He, K., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2961–2969. IEEE, Piscataway, NJ (2017)
41. Zhao, H., et al.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2881–2890. IEEE, Piscataway, NJ (2017)
42. karras, t., et al.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8110–8119. IEEE, Piscataway, NJ (2020)
43. Liu, Z., et al.: A ConvNet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11976–11986. IEEE, Piscataway, NJ (2022)
44. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Piscataway, NJ (2016)
45. Ng, A.Y.: Feature selection, L1 vs. L2 regularization, and rotational invariance. Proceedings of the Twenty-First International Conference on Machine Learning, pp. 770–771. IEEE, Piscataway, NJ (2004)
46. Fortunato, M., et al.: Noisy networks for exploration. arXiv:1706.10295, (2017)
47. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv:1412.6980, (2014)
48. Agustsson, E., Radu, T.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1122–1131. IEEE, Piscataway, NJ (2017)

49. Kishore, V., et al.: Fixed neural network steganography: Train the images, not the network. In: Proceedings of the 9th International Conference on Learning Representations (2021)
50. Luo, Z., et al.: Securing fixed neural network steganography. In: *Proceedings of the 31st ACM International Conference on MultimediaAssociation for Computing Machinery*, pp. 7943–7951. Association for Computing Machinery, New York, NY (2023)
51. Qin, J., et al.: Coverless image steganography based on generative adversarial network. Mathematics 8(9), 1394 (2020)
52. Chen, X., Kishore, V., Weinberger, K.Q.: Learning iterative neural optimizers for image steganography. arXiv:2303.16206, (2022)
53. Benedikt, B.: Stegexpose-A tool for detecting LSB steganography. arXiv:1410.6656, (2014)
54. Fridrich, J., Miroslav, G., Rui, D.: Reliable detection of LSB steganography in color and grayscale images, In: *Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, pp. 27–30. ACM, New York, NY (2001)
55. Dumitrescu, S., Xiaolin, W., Zhe, W.: Detection of LSB steganography via sample pair analysis. Signal Process. IEEE Trans. 51(7), 1995–2007 (2003)
56. Dumitrescu, S., Xiaolin, W., Nasir, M.: On steganalysis of random LSB embedding in continuous-tone images. In: Proceedings of the 2002 International Conference on Image Processing, vol. 3, pp. 641–644. IEEE, Piscataway, NJ (2002)
57. Wu, J., et al.: A coverless information hiding algorithm based on grayscale gradient co-occurrence matrix. IETE Tech. Rev. 35(1), 23–33 (2018)
58. Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of JPEG images: Breaking the F5 algorithm. In: Proceedings of the 5th International Workshop on Information Hiding, pp 310–323. Springer, Cham (2003)
59. Chen, Z., et al.: Invertible mosaic image hiding network for very large capacity image steganography. In: Proceedings of the ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4520–4524. IEEE, Piscataway, NJ (2024)
60. Huang, C.-H., Wu, J.-L.: Image data hiding in neural compressed latent representations. In: 2023 IEEE International Conference on Visual Communications and Image Processing (VCIP), pp. 1–5. IEEE, Piscataway, NJ (2023)
61. Gu, W., et al.: Anti-screenshot watermarking algorithm for archival image based on deep learning model. Entropy 25(2), 288 (2023)

How to cite this article: Ramandi, V.Y., Fateh, M., Rezvani, M.: VidaGAN: Adaptive GAN for image steganography. IET Image Process. 18, 3329–3342 (2024). https://doi.org/10.1049/ipr2.13177