Received: 18 January 2022 | Revised: 19 July 2022 | Accepted: 4 October 2022

DOI: 10.1002/widm.1481

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Deep learning based image steganography: A review

Mohd Arif Wani | **Bisma Sultan**

Department of Computer Science, University of Kashmir, Srinagar, Jammu and Kashmir, India

Correspondence
Mohd Arif Wani, Department of Computer Science, University of Kashmir, Srinagar, Jammu and Kashmir, India.
Email: awani@uok.edu.in

Edited by: Mehmed Kantardzic, Associate Editor and Witold Pedrycz, Editor-in-Chief

Abstract

A review of the deep learning based image steganography techniques is presented in this paper. For completeness, the recent traditional steganography techniques are also discussed briefly. The three key parameters (security, embedding capacity, and invisibility) for measuring the quality of an image steganographic technique are described. Various steganography techniques, with emphasis on the above three key parameters, are reviewed. The steganography techniques are classified here into three main categories: Traditional, Hybrid, and fully Deep Learning. The hybrid techniques are further divided into three sub-categories: Cover Generation, Distortion Learning, and Adversarial Embedding. The fully Deep Learning techniques, based on the nature of the input, are further divided into three sub-categories: GAN Embedding, Embedding Less, and Category Label. The main ideas of the important deep learning based steganography techniques are described. The strong and weak features of these techniques are outlined. The results reported by researchers on benchmark data sets CelebA, Bossbase, PASCAL-VOC12, CIFAR-100, ImageNet, and USC-SIPI are used to evaluate the performance of various steganography techniques. Analysis of the results shows that there is scope for new suitable deep learning architectures that can improve the capacity and invisibility of image steganography.

This article is categorized under:

Technologies > Computational Intelligence
Technologies > Machine Learning
Technologies > Artificial Intelligence

KEYWORDS

deep learning based steganography, GAN-based steganography, image steganography, image steganography classification, image steganography evaluation, steganalysis

1 | INTRODUCTION

The two information hiding techniques of cryptography and steganography are widely used for communicating secret information securely. Cryptography uses a key and an encryption algorithm to convert the secret information into a form called cipher data that is difficult to comprehend. This cipher data is converted back to the original format at the receiver end by using a decryption algorithm. Another way of hiding secret information securely is through steganography, which embeds the secret data in a cover medium. Thus, in cryptography, it is known that communication of secret

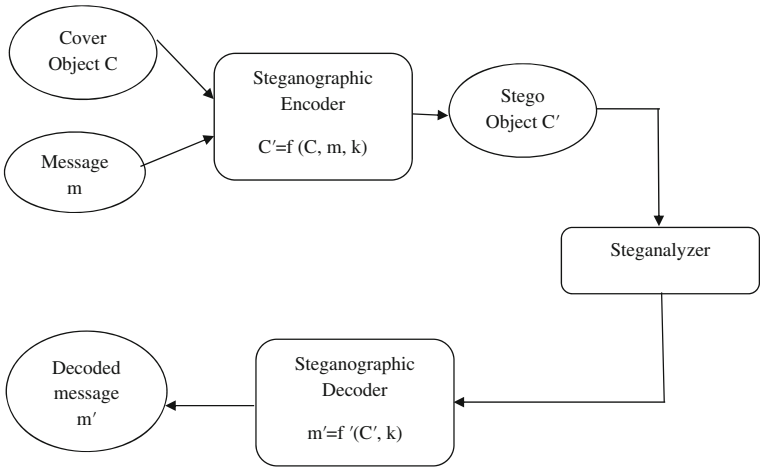

FIGURE 1 General layout of a typical steganography process

information is taking place whereas in steganography it is not known if a cover medium has secret information embedded in it or not.

The secret information can be text, an image, audio, or video. Similarly, the cover medium can be text, an image, audio, or video. The general layout of a typical steganography process is depicted in Figure 1.

The steganographic encoder receives the cover object C (image, text, audio, or video) and the secret message m (image, text, audio, or video) to produce a stego object (C') using key k. The steganographic encoder applies the encoding algorithm or function (f) onto the cover C and m to generate a stego object.

$$C' = f(C, m, k).$$

This stego object is transmitted through a communication channel to the receiver, which acts as a steganographic decoder and extracts decoded secret message (m') from the stego object using key k. The steganographic decoder applies the decoding algorithm or function (f') onto the stego object C' to obtain the decoded secret message m'. The decoder is said to be successful if it can extract m' such that $m = m'$.

$$m' = f'(C', k).$$

During transmission of the stego object from the encoder to the decoder, the steganalyzer analyses the stego object. This process of intrusion into the communication channel to find out whether the secret communication is taking place by the intruder is known as steganalysis. The security of a steganographic system can be increased by making the probability distribution $P_{C'}$ of the stego object similar to the probability distribution P_C of the cover object. Thus

$$D(P_{C'}, P_C) < \in,$$

where D is the distance measure.

Several papers are available in the literature that discusses image steganography techniques. This paper reviews and analyzes various image steganography techniques. The paper is organized as follows: Section 2 describes evaluation criteria and metrics for image steganographic techniques. An overview of image steganography techniques is presented in Section 3. Section 4 discusses the classification of image steganographic techniques. A description of deep learning based image steganography techniques is presented in Section 5. Performance comparison is discussed in Section 6. The conclusion is finally presented in Section 7.

2 | EVALUATION OF IMAGE STEGANOGRAPHIC TECHNIQUES

Selecting an appropriate evaluation mechanism for analyzing how well a steganographic technique is performing is important. This section describes a mechanism that is used to evaluate steganographic techniques.

2.1 | Criteria for evaluating steganography techniques

The quality of a steganographic technique is characterized by three key parameters: security, hiding capacity, and invisibility (or distortion measure). The steganographic techniques reviewed here are evaluated using these parameters. A definition and metric of these parameters are presented below.

2.1.1 | Security

Security measures resistance offered by a steganography technique to steganalysis test. A steganographic technique is said to be secure if it can hide the secret data inside the image in such a way that steganalysis is not able to detect the presence of secret data inside it. Some popular steganalysis tools are Xu's Net (Xu et al., 2016), Ye's Net (Ye et al., 2017) SRM (Fridrich & Kodovsky, 2012), and ATS (Lerch-Hostalot & Megías, 2016).

2.1.2 | Hiding capacity

Hiding capacity measures the amount of secret information that can be embedded inside an image. With the increase in embedding capacity, the stego image quality decreases; hence the two are inversely proportional to each other.

2.1.3 | Invisibility

Invisibility measures similarity between the cover image and the stego image. The goal of steganographic techniques is to embed the secret data such that it results in stego images that are indistinguishable from the cover images, that is, embedded data is invisible.

2.2 | Evaluation metrics

The metrics of the three parameters: (i) Security, (ii) Hiding Capacity, and (iii) Invisibility for evaluating the performance of a steganographic technique are given below.

2.2.1 | Security

Steganalysis test determines whether a given image is embedded with secret data or not; it labels a given image as a cover or stego. A high error rate implies a secure steganographic technique and is computed using the following: number of true positives (TP), number of false negatives (FN), number of true negatives (TN), and number of false positives (FP). The accuracy and error rate are defined below:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

$$\text{Error rate} = (1 - \text{Accuracy}) * 100.$$

2.2.2 | Hiding capacity

Hiding capacity is the number of secret data bits that can be effectively embedded inside the cover image. It is computed as the ratio of maximum hiding capacity to the image size.

$$\text{Relative capacity} = \frac{\text{Absolute capacity}}{\text{Image size}}.$$

Maximum hiding capacity or absolute capacity gives the number of secret data bits hidden inside an image and relative capacity gives the maximum number of bits hidden per pixel. Relative capacity is also referred to as bit rate, and is measured in bits per pixel (bpp) or bytes per pixel (Bpp).

2.2.3 | Invisibility (or distortion measurement)

Various metrics can be used to measure the distortion introduced in a stego image as a result of secret information embedding. These include Mean Square Error (MSE), Root Mean Square Error (RMSE), Correlation, Quality Index, Peak Signal-to-Noise Ratio (PSNR), Weighted PSNR (WPSNR), Structural SIMilarity (SSIM), Manhattan Distance, Euclidean Distance and Kullback–Leibler Divergence (K-L divergence). PSNR and SSIM are the two commonly used metrics for distortion measurement in steganography. A high value of PSNR and SSIM denote better image quality and lesser distortion and vice versa.

PSNR is defined as the ratio of the maximum power to the power of noise that influences the quality of an image

$$\text{PSNR} = 10\log_{10}\left(\frac{(\text{Max}_I)^2}{\text{MSE}}\right).$$

Here, Max_I indicates the maximum possible intensity levels (pixel values) in an image, and MSE is the mean square error that is given as:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(C(i,j) - C\prime(i,j))^2,$$

where $C(i, j)$ represents the cover image pixel and $C'(i, j)$ represents the stego image pixel. m and n represent the number of rows and columns, respectively, in the images.

SSIM measures the similarity between two images and is given as:

$$\text{SSIM}(C, C') = \frac{(2\mu_c\mu_{c'} + C1)(2\sigma_{cc'} + C2)}{(\mu_c{}^2 + \mu_{c'}{}^2 + C1)(\sigma_c{}^2 + \sigma_{c'}{}^2 + C2)}.$$

Here, μ_c denotes the mean intensity of cover image (C), and $\mu_{c'}$ denotes the mean intensity of the stego image (C'); $\sigma_c{}^2$ and $\sigma_{c'}{}^2$ represent the variance of C and C', respectively; $\sigma_{cc'}$ gives the covariance between C and C'; $C1$ and $C2$ are two parameters for stabilizing weak denominator divisions and are given by $C1 = (k_1\ L)^2$ and $C2 = (k_2\ L)^2$, L indicates the dynamic range of pixels values (typically this is $2^{\#\ \text{bits per pixel}} - 1$) and k_1 and k_2 are set to 0.01 and 0.03, respectively, as default values.

3 | PREVIEW OF IMAGE STEGANOGRAPHY TECHNIQUES

The recent image steganography approaches that have improved security, capacity, and imperceptibility are reviewed in this section.

3.1 | Steganography techniques with emphasis on security

This subsection reviews image steganographic techniques that attempt to improve security. The salient features of these techniques are reviewed separately under traditional, hybrid, and fully deep learning sub-headings.

3.1.1 | Security of traditional techniques

Several traditional techniques have been proposed in the literature for improving the security of steganography. These techniques use human-designed rules for embedding secret information in the cover medium. Some of these techniques use the spatial domain cover medium and other techniques use the transform domain cover medium. In the spatial domain, image pixels are directly manipulated, whereas, in the transform domain, manipulation is done on frequency values. One of the well-known spatial domain techniques is called least significant bit (LSB) substitution (Mielikainen, 2006). The technique uses a subset of cover images (C_1, C_2, C_3, ..., C_i) and replaces the least significant bit of the cover image pixels with the secret information bit (which can be either 0 or 1). On the receiver side, the extraction algorithm extracts the secret information bits from the image and reconstructs the secret message. Variants of the LSB technique have been designed to increase security. A chaotic sequence is generated by authors in Rajendran & Doraipandian (2017) by using a 1-D logistic map and then hiding the secret data based on this chaotic sequence.

Content-based image steganography employs a distortion function to find out those areas in an image where the secret data can be embedded with less distortion. Content-based techniques such as HUGO (Pevný et al., 2010), WOW (Holub & Fridrich, 2012), and S-UNIWARD (Holub et al., 2014) are secure and widely used spatial domain image steganography techniques. Authors in Liao et al. (2020) design a channel-dependent payload partition scheme that is based on Amplifying Channel Modification Probabilities (ACMP), so as to adaptively assign embedding capacity among RGB channels. The method embeds the secret data in the heavily textured regions of R B and G channels as per the adaptively assigned embedding capacity scheme which improves the security.

Authors in Qu et al. (2019) use Exploitation Modification Direction (EMD) technique to hide the data in quantum carrier images for improved security. The method described in Lu et al. (2021) uses a pixel density histogram (PDH) of a halftone image to select the blocks with complex textures for data embedding. The data is embedded in the selected blocks using LSB substitution. The method in Li & Zhang (2019) uses steganography without embedding that constructs a fingerprint directly from the secret message. Authors in Liao et al. (2022) increase steganography security by distributing the embedding payload over multiple images adaptively making it difficult for the steganalyzer to identify the stego from a sequence of images. The payload is distributed into multiple images based on image texture complexity and distortion distribution strategy.

3.1.2 | Security of hybrid techniques

Generative Adversarial Networks (GANs) (Goodfellow et al., 2020) have been combined with traditional techniques in Volkhonskiy et al. (2020) for the steganography process. The process uses DCGAN for generating cover images and the secret information is embedded using the LSB embedding algorithm. Here, SGAN is used to train the Generator against the Discriminator and the Steganalyzer. This simultaneous training helps in generating secure cover for steganography as these images are less susceptible to steganalysis. To speed up and stabilize the training and to enhance the quality and security of the generated images, a model called SSGAN is proposed in Shi et al. (2018). The model estimates the adaptability of generated images by employing GNCNN (Qian et al., 2015) as the Steganalyzer network.

Some researchers have incorporated GANs to automatically learn the distortion function for data hiding. Authors in Tang et al. (2017) developed ASDL-GAN to learn the distortion function automatically. A Model called UT-SCA-GAN (Yang et al., 2018) incorporates a U-NET (Navab et al., 2015) based Generator to generate the probability map. The generated probability map combined with uniform random noise is fed into the Tanh simulator for generating a modified map. The modified map and the cover image are used to generate the stego image. Authors in Yang et al. (2020) use UNET as a generator and XuNET with six high pass filters as a discriminator. The use of multiple high pass filters results in a better steganographic scheme as reported in Ye et al. (2017), Yedroudj, Comby, & Chaumont (2018) and Yedroudj, Chaumont, & Comby (2018).

Authors in Zhang et al. (2018) add adversarial noise to cover images to generate adversarial examples by using the Fast Gradient Sign based technique (Goodfellow et al., 2015). The secret message is then embedded in the generated adversarial images. The concept of using adversarial examples with adversarial training is used in Zhou et al. (2020) to improve security. The authors in Ma et al. (2019) use an adversarial technique in distortion minimizing steganography to improve security. The authors in Tang et al. (2019) use an adversarial embedding scheme in steganography to improve security.

3.1.3 | Security of fully deep learning techniques

Several researchers have explored using fully deep learning based image steganography. This approach uses GANs for the entire steganographic process and no traditional embedding techniques are used. Authors in Shi et al. (2019) have developed a model in which the discriminator is responsible not only for the realism of the stego images but also acts as an extractor network with the help of a decoding function. The steganalyzer network is utilized to distinguish between cover and stego images. Discriminator and Steganalyzer are trained adversarially with Generator so that each network improves resulting in better security. Authors in Yedroudj et al. (2020) have introduced two different keys for Generator and Extractor networks for better security. AHIGAN model has been developed by Fu et al. (2020), where the encoder-decoder network and XuNET (Discriminator/steganalyzer) are employed to achieve the task of hiding a color image inside another color image with strong resistance to steganalysis.

Steganography without embedding (SWE) technique or embedding less technique has been explored by researchers where no explicit embedding is used. It uses noise and secret information to generate stego images with GANs. Authors in Wang et al. (2018) have developed an SWE model where noise and secret messages are fed into the Generator as input. The stego image is generated by performing adversarial training of GAN. To improve the security of the SWE technique, authors in Hu et al. (2018) have created a relationship between noise and the secret message, and then the modified noise vector is fed into the Generator for generating stego images. A Secure Generative Reversible Data Hiding (GRDH) model has been developed by Zhang, Fu, et al. (2019) for an image-to-image translation using cyclic GAN. The model uses the concept of the encrypted image described in Zhang et al. (2016) and the secret message mapping technique of Hu et al. (2018). Authors in Ke et al. (2017) proposed generative steganography based on Kirchhoff's law. Two GANs, namely Cover-GAN and Message-GAN, work together to generate a secure stego image while maintaining Kirchhoff's law instead of embedding it into the cover image.

Some researchers have explored the Category Label technique for steganography. This technique replaces labels with secret data. The labels are class labels of the samples in a dataset and act as a driver to generate a steganographic image, which improves security. Authors in Zhang et al. (2020) have designed a Synthetic Semantic Stego Generative Adversarial Network (SSS-GAN), which creates a mapping relationship between the secret information and semantic labels of the images. These semantic labels are then used along with a random noise vector to generate the stego images. On the receiver side, the secret message is extracted from the semantic labels of the images by reverse mapping rules. This mechanism makes steganography more secure. Authors in Liu, Zhou, et al. (2018) have used ACGAN for coverless image steganography. The labels in the ACGAN Generator are replaced with secret information using a code dictionary. The relationship between the labels and secret data is created using the mapping function provided by the authors in Hu et al. (2018). The stego image is generated by trained ACGAN using noise and secret fragments.

Table 1 lists some important steganography techniques for improving security. This includes traditional techniques like distortion minimization, LSB, and fingerprint steganography; hybrid techniques that use deep learning modules like GAN, WGAN, Pixel CNN, and encoder-decoder; and fully deep learning techniques that use real covers or noise based mapping or label based mapping for adversarial training for improving security.

3.2 | Steganography techniques with emphasis on capacity

The recent traditional and deep learning approaches that have been employed by various researchers to increase the embedding capacity of a steganography system are reviewed here.

3.2.1 | Embedding capacity of traditional techniques

Embedding capacity, also called embedding payload, represents the percentage of embedded secret bits in the whole pixels of the cover image. Different algorithms have been proposed by researchers to increase the embedding capacity in steganography. Authors in Lu et al. (2015) increase the capacity of the LSB technique by extending it to four LSB planes, but this increases distortion. The authors in Swain (2019) employ a two-level embedding strategy using LSB and quotient value differencing (QVD) techniques to improve capacity. Another technique known as pixel value differencing (PVD) employs the difference between the two consecutive pixels to determine the capacity for hiding the secret data. Various PVD techniques have been introduced depending on the number of neighboring pixels used for

TABLE 1 Steganography methods with emphasis on security

Method/references	Embedding technique	Key features	Method type	Remarks
Pevný et al. (2010)	Distortion minimization	Defines a distortion function, and assigns cost to pixels based on the effect of embedding some information within a pixel.	Traditional	Limited capacity
Holub and Fridrich (2012)	Distortion minimization	Employs WOW that embeds the secret data in complex areas of an image resulting in good security.	Traditional	Limited capacity
Lu et al. (2021)	Least Significant Bit (LSB) variant	Uses pixel density histogram (PDH) of a halftone image to select the blocks with complex textures.	Traditional	Limited capacity
Li & Zhang (2019)	Fingerprint steganography	Constructs a fingerprint directly from a secret message. The continuous and spiral phases in the fingerprint result in secure steganography.	Traditional	Limited capacity
Tang et al. (2017)	Syndrome-Trellis Code (STC)	Employs GAN (ASDL-GAN) to learn the probability of data embedding areas.	Hybrid	Limited capacity
Yang et al. (2020)	Syndrome-Trellis Code (STC)	Employs U-NET based generator and six high pass filters based discriminator to improve security.	Hybrid	–
Volkhonskiy et al. (2020)	Least Significant Bit (LSB), Highly Undetectable stego (HUGO)	Employs GAN(SGAN) to generate cover images. Embeds data in cover images using LSB or HUGO algorithm.	Hybrid	Low invisibility and capacity
Zhang et al. (2018)	Wavelet Obtained Weights (WOW)	Generates adversarial images to be used as cover images. Embeds data in cover images using the WOW algorithm	Hybrid	–
Fu et al. (2020)	Employs real covers and secret data for adversarial training	Uses encoder-decoder based architecture with residual and skip connections to generate secure stego images.	Fully deep learning	–
Hu et al. (2018)	Employs noise-based mapping for adversarial training	Generates stego images from modified noise vector.	Fully deep learning	Complex approach
Zhang et al. (2020)	Employs label-based mapping for adversarial training	Employs Synthetic Semantics GAN to generate stego images directly from noise and labels.	Fully deep learning	Limited capacity

enhancing the embedding capacity (Swain & Lenka, 2015). Authors in Grajeda-Marín et al. (2018) have introduced optimal tri-way PVD to resolve the falling-off boundary issue. An approach that combines PVD with LSB has been used by authors in Swain (2016), where the image is divided into small patches, and then the secret data is hidden in these small patches using the LSB technique. Parity-bit pixel value difference PBPVD technique (Hussain et al., 2017) employs PVD in order to embed an extra parity-bit in each block without affecting visual quality. PVD techniques achieve high embedding capacity, but the main disadvantage is the lack of resistance to steganalysis. Histogram Shifting (HS) based steganography has been combined with bit-plane slicing in Nyeem (2018) to enhance the embedding capacity.

A high-capacity 3D steganography model has been developed by the authors in Li et al. (2017). Truncated space has been utilized along with shifting policy to effectively embed the data in the 3D model. A DCT steganography technique has been designed by Rabie & Kamel (2017) that utilizes a global-adaptive-region (GAR) scheme to increase the embedding capacity. The technique explores the likeness of the cover image DCT coefficients and secret image data and hides the data in the most correlated DCT block.

3.2.2 | Embedding capacity of fully deep learning techniques

The use of deep learning at first mainly focused on improving the security of the steganography process. With the success in designing secure steganography systems, research is being carried out to increase the capacity of deep learning-based steganography. Some researchers have explored GAN embedding techniques for improving steganography capacity. The authors in Baluja (2017) employ a deep neural network for embedding a color image into another color image of the same size. The model consists of three networks: a preparation network, which makes the size of the secret image the same as that of the cover image; the second network is a hiding network that generates the stego image; the last network is the reveal network for extraction of the secret image. Authors in Li et al. (2018) use two approaches to improve capacity: the first approach embeds the secret message by a conceal network into a complex texture-like image generated by DCGAN; the second approach combines the conceal network and DCGAN into one network for generating a complex texture-like image for embedding the secret image. The authors in ur Rehman et al. (2019) introduce a new loss function, which is a combination of encoder and decoder loss, for improving the embedding capacity.

The model described in Wu et al. (2018a) is a high-capacity image steganography model developed without using a traditional rule-based algorithm. The secret image and cover image are concatenated and fed into an encoder network which translates the high-level features of the input image into latent representation and transforms it into an image similar to the cover image. A SteganoGAN model has been introduced by Zhang, Cuesta-Infante, et al. (2019) that increases the payload of GAN-based steganography10 times by using the encoder, decoder, and critic network with dense and residual connections. The model described in Duan et al. (2019) employs U-NET as its hiding network that distributes the compressed secret image bits over the cover image to obtain the stego image, which is then decoded to get back the secret image using an extractor network.

Table 2 lists some of the recently designed steganography methods for improving the embedding capacity. This includes traditional techniques like PVD, PBPVD, LSB + PVD, LSB + QVD, and bit plane slicing with histogram shifting; fully deep learning models with different architectures (encoder, decoder, U-NET, etc.) and loss functions for improving the embedding capacity.

3.3 | Steganography techniques with emphasis on invisibility

Recent traditional and deep learning approaches that have improved the invisibility or imperceptibility of the image steganography are reviewed in this section.

3.3.1 | Invisibility of traditional techniques

The visual quality of a reversible data hiding scheme is improved in Chang et al. (2017) by dividing the image pixels into embeddable and non-embeddable pixels. Data is then stored in the embeddable pixels which result in minimal distortion, thus improving invisibility. The technique in Luo et al. (2018) increased the visual quality of the reversible data hiding (RDH) scheme by utilizing just noticeable difference (JND) technique. Authors in Mandal et al. (2017) designed steganography technique using Bit Plane Complexity Segmentation (BPCS) and Hessenberg QR techniques. A minimal distortion steganography technique using DWT is proposed in Kumar & Kumar (2018). It uses secret key computation and blocking concepts to improve the visual quality of imperceptibility. The authors in Miri & Faez (2018) store the secret data in the highest frequency (HH) matrix of the integer wavelet transform of the cover image. This embedding is based on the fact that human eyes are not sensitive to changes in the high-frequency areas of edges.

3.3.2 | Invisibility of fully deep learning techniques

The GAN embedding deep learning approach has been used by some researchers to improve the invisibility of steganography. Authors in Zhang, Dong, & Liu (2019) propose an architecture, namely ISGAN, that is capable of hiding a secret gray image inside a color image. The model does not use the two channels Cr and Cb which store the color information for embedding but hides the secret image only in the Y channel of the cover image resulting in better

TABLE 2 Steganography methods with emphasis on capacity

Method/references	Embedding technique	Key features	Method type	Remarks
Grajeda-Marín et al. (2018)	Pixel value differencing (PVD)	Searches the best value for each pixel pair such that no blocks of pixels are left out of embedding.	Traditional	Poor security and visual quality
Hussain et al. (2017)	Parity-bit pixel value difference (PBPVD)	Adjusts additional secret data by using an extra bit for embedding.	Traditional	–
Swain (2016)	LSB + PVD	Increases capacity by combining LSB with PVD	Traditional	Poor Security
Swain (2019)	LSB + QVD	Increases capacity by combining LSB with QVD	Traditional	Fall off boundary issue
Nyeem (2018)	Bit plane slicing with histogram shifting	Divides an image into two low-intensity images, data is stored in these individual images which increases capacity	Traditional	Poor security
Zhang, Dong, & Liu (2019)	Uses generator with adversarial training	Employs GAN with encoder-decoder based network with dense and residual connections to increase the payload	Fully deep learning	–
ur Rehman et al. (2019)	Uses encoder decoder with adversarial training	Employs encoder-decoder based architecture with a new loss function to generate high capacity stego images	Fully deep learning	
Duan et al. (2019)	Uses U-Net for adversarial training	Employs U-NET based generator that compresses and distributes the secret image over the cover image for improved capacity	Fully deep learning	–
Wu et al. (2018b)	Uses real covers for adversarial training	Employs an encoder network that translates the input image into latent representation and transforms it into an image similar to the cover image.	Fully deep learning	The quality of embedded images is compromised

invisibility. Authors in Ahuja et al. (2019) develop a color image steganography model with good imperceptibility by performing steganography and steganalysis simultaneously using GAN.

A Hidden model has been proposed by Zhu et al. (2018) that performs both steganography as well as watermarking using Deep Neural Networks. Authors in Huang et al. (2019) introduce a new loss function that helps to store the secret information in complex texture features of an image. A discriminative model and inconsistency loss are used in With (2019) to improve the imperceptibility of the color image steganography. Authors in Sultan & Wani (2021) show that secret data embedded in XYZ image format provides better invisibility than RGB images for deep learning steganography. Authors in Tan et al. (2022) propose a GAN based steganography with channel attention mechanisms for improved visibility at various embedding capacities. The method can learn channel interdependencies and adaptively adjust channel-wise features in the network activation of images, through which it improves the quality of generated stego images.

Table 3 lists some of the recently devised steganography methods for improving invisibility. This includes employing traditional techniques like GAP, Hessenberg QR decomposition, CRT, and CRT + PVD; fully deep learning models with different architectures like encoder-decoder, residual blocks and strategies like using only one channel of cover for embedding for improving invisibility.

4 | CLASSIFICATION OF IMAGE STEGANOGRAPHIC TECHNIQUES

Image Steganographic techniques are broadly classified into three categories based on the embedding strategy used. The three categories are: Traditional techniques, Hybrid techniques, and fully Deep Learning techniques as shown in Figure 2, and are briefly discussed below.

TABLE 3 Steganography methods with emphasis on invisibility

Model/references	Embedding technique	Key features	Method type	Remarks
Chang et al. (2017)	Gradient-adjusted prediction (GAP)	The method divides the image pixels into embeddable and non-embeddable pixels. The secret data is embedded in the embeddable pixels using the GAP method.	Traditional	–
Mandal et al. (2017)	Hessenberg QR decomposition	The technique uses Bit Plane Complexity Segmentation (BPCS) to divide the cover image into bit planes and selects the noisy bit plane for data embedding.	Traditional	–
Liao et al. (2018)	CRT. CRT + PVD	Employs cubic reference table (CRT) which increases the dimensionality of reference table from 2 to 3. Combines PVD with CRT to further improve visual quality.	Traditional	As capacity increases, the image quality decreases
Ahuja et al. (2019)	Uses real covers for adversarial training	Introduced cycle consistency loss and inconsistent loss to increase the invisibility of the steganography process.	Fully deep learning	–
Zhang, Dong, & Liu (2019)	Uses one channel of cover for adversarial training	The model converts the RGB carrier image into a YCrCb image and hides the secret image only in the Y channel. The Cr and Cb channels contain color information, and are not used for embedding.	Fully deep learning	–
Huang et al. (2019)	Uses residual blocks in the generator for adversarial training	Stores the secret information in complex texture features of an image.	Fully deep learning	As the capacity increases, distortion increases.

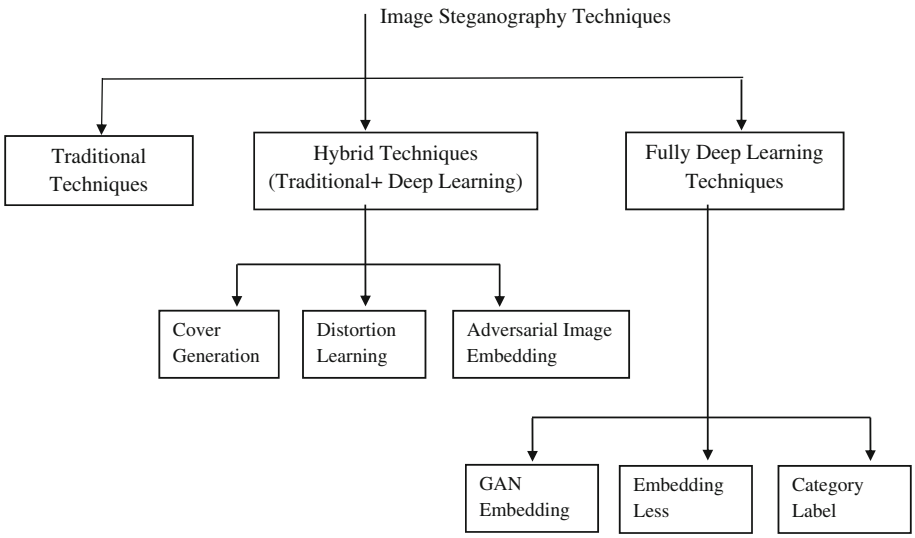

FIGURE 2 Classification of image steganography techniques

4.1 | Traditional techniques

Traditional techniques use rules framed by humans for encoding and decoding secret information. A large number of such steganography techniques, to improve security, capacity, and invisibility, have been proposed in the literature. Some of these techniques have been reviewed in the previous section, and a comparison of the performance of these steganography techniques is given in Section 6.

4.2 | Hybrid techniques

A steganographic process where a part of the process is completed using rules framed by humans and a part of the process is performed by a trainable module falls under the category of hybrid approach. Deep learning techniques have been used in the field of steganography as trainable modules. The most commonly used deep learning module in steganography is a deep neural network or a GAN. Traditional techniques have been combined with a deep neural network or a GAN for designing hybrid systems for improving steganography. The hybrid techniques are further classified into three sub-categories viz; Cover Generation, Distortion Learning, and Adversarial Image Embedding.

(a) Cover Generation:

Instead of using the real images for steganography, artificial images generated by GAN models like DCGAN, and WGAN are used for secret information hiding. Embedding is performed using the traditional algorithms such as LSB, HUGO, WOW, and UNIWARD. The images generated by GANs are less susceptible to steganalysis.

(b) Distortion Learning

Distortion learning techniques use high pass filtered cover images in distortion generators to produce distortion maps for steganography. GAN-based distortion learning models are commonly used to generate distortion maps. A pre-trained module is used to embed secret messages in the distortion map.

(c) Adversarial Image Embedding

This category of steganography techniques hides secret data in adversarial images by employing the traditional embedding techniques such as WOW, S-UNIWARD, and LSB. The use of an adversarial image for steganography improves security.

4.3 | Fully deep learning techniques

Fully Deep Learning Techniques have been proposed to develop unsupervised steganographic systems that require no rules framed by humans. These techniques do not use any traditional embedding algorithms for data hiding. Instead, the network is trained in an unsupervised manner to learn the full steganography process. The techniques falling under this category are divided into three sub-categories: GAN Embedding, Embedding Less, and Category Label.

(a) GAN embedding

GAN embedding techniques use cover image and secret data as input to the GAN Generator, which embeds the secret message inside the cover image. The GAN automatically learns the embedding process without using the traditional steganographic algorithms.

(b) Embedding less

Embedding less techniques employ a steganography process that does not require an explicit embedding step. These techniques are also called coverless techniques, as no cover images are required for the secret data. The stego image is directly generated by the model by using noise and the secret message as input.

(c) Category label

Category Label techniques map secret image chunks to image labels as one of the important steps of the steganographic process. Noise and image labels are used as input to the generator for the generation of stego images.

5 | DEEP LEARNING BASED STEGANOGRAPHY TECHNIQUES

This section discusses the important points of deep learning based steganography techniques. The workflow diagrams of these techniques are discussed. The strong and weak features are discussed briefly at the end of this section.

5.1 | Cover generation techniques

Cover generation techniques generate secure cover images for steganography. These artificial cover images are generated by GAN models like SGAN, and WGAN. The techniques described in SGAN Volkhonskiy et al. (2020), Zi et al.

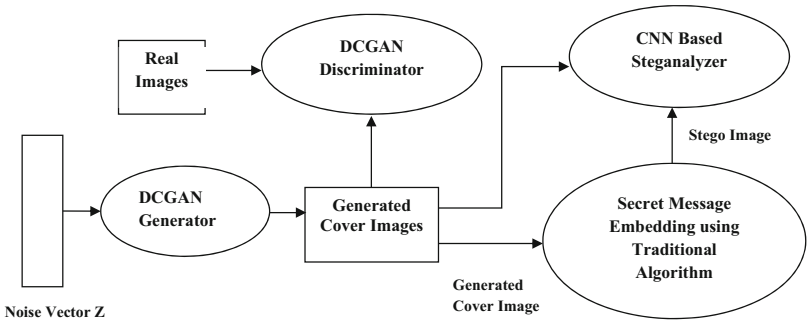

FIGURE 3 Workflow diagram of SGAN

(2019) and Shi et al. (2018) have been developed for generating secure covers for steganography, and a comparison of the performance of these models is given in Table 5. The workflow diagram of the SGAN model for image steganography is shown in Figure 3.

The model consists of a DCGAN generator, a DCGAN discriminator, and a CNN-based steganalyzer. DCGAN generator receives random noise as input and generates an image from this noise. The discriminator receives generated and real images alternatively and its role is to distinguish between real and generated images. A secret message is embedded in the generated cover image using a traditional steganography algorithm like LSB. In the original DCGAN, only the discriminator is used to perform classification, but in SGAN, steganalyzer is also used to perform binary classification. Steganalyzer's role is to distinguish between the generated cover image and the stego image. During the early stages of training, the generator produces images that do not resemble real images. As the training progresses, the generator produces better and better images with the help of the feedback received from the discriminator and the steganalyzer. The trained generator produces cover images that cannot be differentiated from the real mages or stego images. This results in secure cover images for steganography. The generator is trained against the discriminator and the steganalyzer simultaneously using the loss function given below:

$$L = \alpha\left(E_{X\sim p(\boldsymbol{X})}[\log D(X)] + E_{Z\sim p(\boldsymbol{Z})}[\log(1 - D(G(Z)))]\right) + (1-\alpha)E_{Z\sim p(\boldsymbol{Z})}[\log S(\text{Stego}(G(Z))) + \log(1 - S(G(Z)))],$$

where α is a parameter that is used to control the trade-off between the realism of covers and security of covers; Z is a noise vector; X is a real image; E (.) is the expectation value; Stego (.) is the output of a traditional embedding algorithm; S (.) is the output of the steganalyzer network; G (.) is the output of the generator network (generated cover image); D (.) is the output of the discriminator network.

5.2 | Distortion learning techniques

Distortion learning techniques use high pass filtered cover images in distortion generators to produce distortion maps for steganography. A pre-trained simulator is used to embed secret messages in the distortion map. The distortion maps are generated by GAN-based distortion learning models like ASDL-GAN (Tang et al., 2017), UT-SCA-GAN (Yang et al., 2018), and UT-6-HPF-GAN (Yang et al., 2020). A comparison of performance is presented in Table 6. The workflow diagram of the ASDL-GAN model for image steganography is shown in Figure 4.

The distortion-based steganography algorithms use a high pass filtered version of the cover image to generate residual maps. On similar lines, GAN-based distortion learning models use high pass filtered versions of cover images for producing distortion maps and in steganalyzer. This ensures consistency between the generator and steganalyzer at the input level. As depicted in Figure 4, the generator network receives the high pass filtered cover image as an input and generates the distortion map. The distortion map is fed into a small network called Ternary Embedding Simulator (TES) that is pre-trained and acts as an activation function for the generation of modified distortion map m. The modified distortion map m and the cover image are added together to form the stego image S. The role of the steganalyzer is

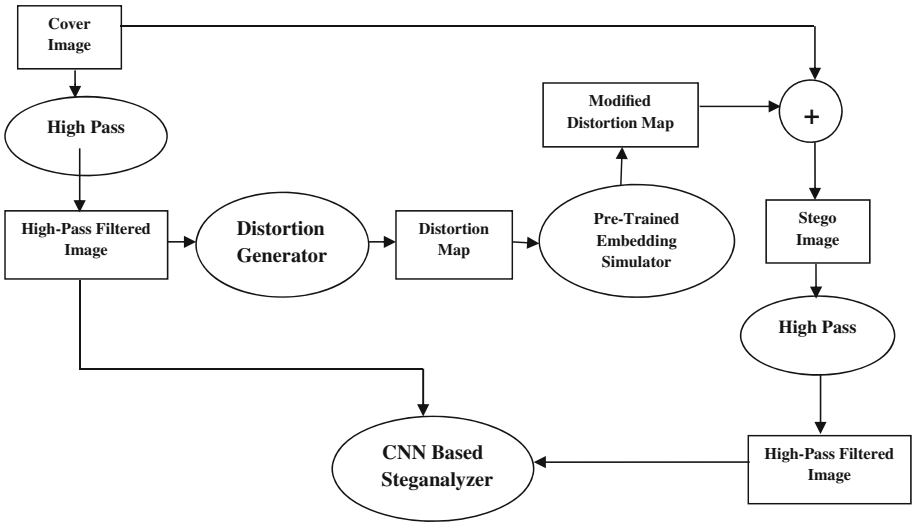

FIGURE 4 Workflow of ASDL GAN.

to distinguish between generated stego images and the cover images. The two networks compete with each other to achieve the task. TES is pre-trained and uses the loss function given below:

$$l_{\text{TES}} = \frac{1}{H}\frac{1}{W}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(m_{i,j} - m'_{i,j}\right)^2,$$

where $m_{i,j}$ is the modified distortion map and $m'_{i,j}$ is the ground truth.

The steganalyzer is trained using the function:

$$l_D = -\sum_{i=1}^{2} y'_i \log(y_i),$$

where y'_i represents ground truth and y_i represents the output of the steganalyzer.

The generator is trained using the loss function given below:

$$l_G = \alpha l_1 + \beta l_2,$$

where α and β are the weightage given to $l_1$and l_2, respectively. l_1 is given as $l_1 = -l_D$ and $l_2 = (\text{Capacity} - H * W * q)^2$. q represents the payload of messages that are carried by the stego image.

5.3 | Adversarial image embedding techniques

Adversarial image embedding techniques utilize adversarial examples for steganography for improved security. The approaches described in Zhang et al. (2018), Zhou et al. (2020), Ma et al. (2019), and Tang et al. (2019) use an adversarial image for the steganography process, and a comparison of performance is given in Table 7. An adversarial example is an image that is obtained by adding adversarial noise to an image. Figure 5 shows the workflow of the adversarial image embedding technique described in Zhou et al. (2020).

As depicted in Figure 5, the original image is fed into the generator network to produce the adversarial noise. The generated adversarial noise is added to the original image to produce the adversarial image. The secret data is then embedded in the adversarial image using the traditional embedding algorithm like WOW and UNIWARD. To improve the security of steganography, CNN based steganalyzer is used that receives adversarial image and stego image alternatively to train the generator so that adversarial and stego images cannot be differentiated. The steganalyzer outputs a normalized two-

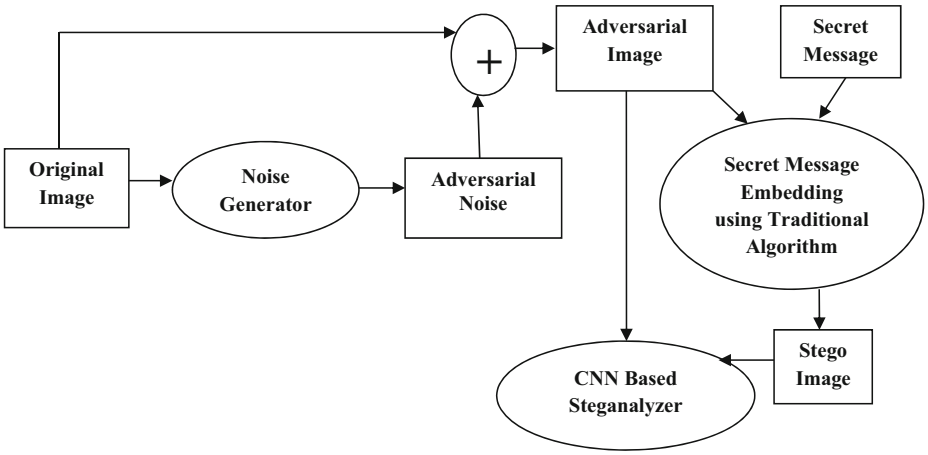

FIGURE 5 Workflow of adversarial embedding technique

dimensional vector. This two-dimensional vector [a, b] determines the final classification result. If $a > b$, the image is more likely to be a cover image, and if $a < b$, the image is more likely to be a stego image. The distance $d = (a - b)$ is used to decide if the image is a cover image or a stego image. The loss function for training the generator network is given as:

$$L_G = \lambda \|X - X'\|_2^2 + f(X')$$

$$f(X') = \text{abs}(Z(X') - Z(\text{Stego}(X'))),$$

where X represents the original image and X' is the adversarial image. λ is the weight parameter of $\|X - X'\|_2^2$ (L2 function); Stego (.) is the output of a traditional embedding algorithm. $Z(X')$ is the distance $d = (a - b)$ when input is a cover adversarial image. $Z(\text{Stego}(X'))$ is the distance $d = (a - b)$ when input is stego image. abs(.) is absolute value.

The steganalyzer is trained using the cross-entropy loss function given below

$$L_S = E[\log(S(X'))] + E[\log(1 - S(\text{Stego}(\text{X}')))],$$

where X' represents the adversarial image. Stego(X') is the stego image produced after embedding secret data in the adversarial image. S(.) represents the output of the steganalyzer network.

5.4 | GAN embedding techniques

GAN embedding techniques for image steganography are unlike traditional algorithms as these techniques are free from any human intervention. A model representing this technique is trained using the adversarial training of GAN to embed a secret message in a cover image. The techniques described in Abadi & Andersen (2016), GSIVAT Hayes & Danezis (2017), Shi et al. (2019), Liu, Zhou, et al. (2018), Yedroudj et al. (2020), Yang et al. (2019), and Fu et al. (2020) are GAN embedding techniques. A comparison of the performance of such techniques is given in Tables 8, 12, and 14. The GAN embedding GSIVAT model workflow is shown in Figure 6 below.

A secret message (M) and a cover image (C) are fed to the generator network to generate the stego image (C′) as shown in Figure 6. The extractor network receives the stego image generated and recovers the secret message. The discriminator network serves as a steganalyzer and receives both the cover image and the stego image alternatively and provides feedback for training the generator. The generator is trained so that the steganalyzer cannot differentiate between the cover image and the stego image. The output value of the steganalyzer that is close to 0 indicates that the input message is a stego image and the value that is close to 1 indicates that the input image is a cover image. Initially, the generator produces noisy images from which the extractor is not able to extract the secret message successfully. As the training progresses, all the networks improve their performances. With the trained generator, the extractor can successfully recover the secret message from the stego image. The extractor network is trained by minimizing the loss function shown below:

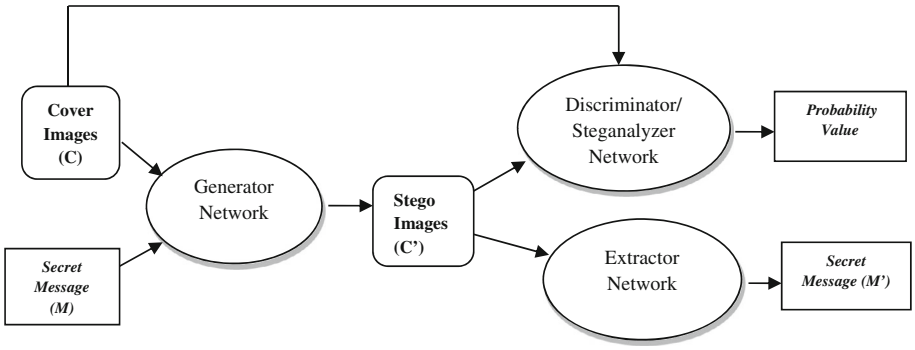

FIGURE 6 Workflow of GSIVAT model

$$L_E(\theta_G, \theta_E, M, C) = d(M, M')$$

$$= d(M, B(\theta_E, C'))$$

$$= d(M, B(\theta_E, A(\theta_G, C, M))),$$

where θ_G and θ_E denote the parameters of the generator and extractor networks; M and M' denote the original secret message and the recovered secret message; C and C' denote the cover and stego images; $d(M, M')$ is the Euclidean distance between M and M'.

The steganalyzer uses the Sigmoid cross-entropy loss function and is given as:

$$L_S(\theta_S, C, C') = -y(\log(E(\theta_S, X))) - (1-y)(\log(1 - E(\theta_S, X))),$$

where θ_S denotes the parameters of the steganalyzer network; $y = 0$ indicates stego image, that is, $X = C'$, and $y = 1$ denotes cover image, that is, $X = C$.

The generator loss function is the weighted sum of the reconstruction loss, the extractor loss and the steganalyzer loss and is given as:

$$L_G(\theta_G, C, M) = \lambda_G d(C, C') + \lambda_E L_E + \lambda_S L_S,$$

where λ_G, λ_E, and λ_S are the weights assigned to the three losses.

5.5 | Embedding less techniques

Embedding less steganography techniques produces a stego image without embedding a secret message in a cover image. Here the model learns how to generate the stego image directly from the noise and secret message. The approaches described in Wang et al. (2018), Zhang, Liu, et al. (2019), Hu et al. (2018), Zhang, Fu, et al. (2019), and Ke et al. (2017) are embedding less approaches. A performance Comparison of these approaches is given in Table 9. The workflow of the embedding less steganography model described in Hu et al. (2018) is shown below in Figure 7.

Figure 7a shows the stego image generation process. The segments S_i of secret information are mapped to the noise vectors Z_i by employing a mapping function. These transformed noise vectors are then fed into the DCGAN generator for the generation of stego images. The trained generator is obtained after DCGAN converges using the loss function given below:

$$\frac{\min}{G}\frac{\max}{D} V(G, D) = E_{X \sim p(\mathbf{X})}[\log D(X)] + E_{Z \sim p(\mathbf{Z})}[1 - \log(1 - D(G(Z)))],$$

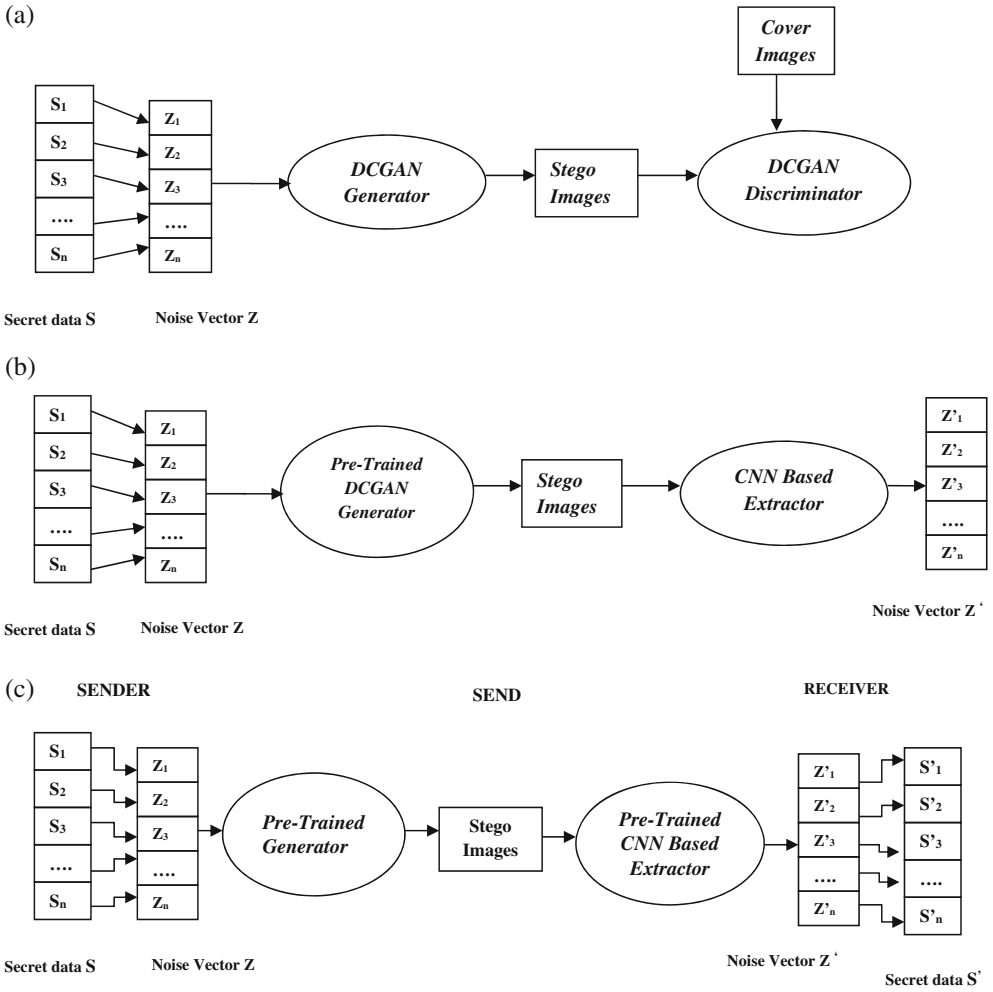

FIGURE 7 (a) Training stego image generation. (b) Training secret message extractor. (c) Workflow of embedding less steganography model

where X represents the cover image; Z is the mapped noise vector; E denotes expectation; $G(Z)$ is the stego image generated by generator G.

Figure 7b displays the training of the extractor network. The extractor network receives the stego image generated by the trained generator and outputs the noise vector. The extractor is trained by using the difference between the original noise vector and the output of the extractor as the recovery accuracy measure. The extractor network training loss function is given as:

$$L_E = \sum_{i-1}^{n} \left(Z - E(\text{Stego})^2\right) = \sum_{i=1}^{n} Z - E(G(Z))^2,$$

where Z is the noise vector, E(Stego) is the noise vector recovered by the extractor network. This noise vector is then used to recover the secret message bits using reverse mapping rules.

Figure 7c displays how the communication takes place. The sender uses the trained generator and the receiver uses the trained extractor. At the sender side, the secret message S is divided into segments S_i and then each segment S_i is mapped into noise vector Z_i, This mapping is fed to the trained generator which generates stego images. The extractor at the receiver side extracts the noise vector from the stego image. The secret data is recovered from the Noise vector as per the reverse mapping rules.

5.6 | Category label techniques

Category label image steganography techniques use the conditional GAN model for the generation of stego images. The category labels and noise are fed to the generator network for the generation of stego images. The labels act as a driver for the stego image generation. The techniques described in Zhang et al. (2020) and Liu, Zhang, et al. (2018) are category label techniques. A comparison of the performance of these techniques is given in Table 10. Figure 8a and 8b show the workflow of the category label image steganography technique described in Liu, Zhang, et al. (2018).

The ACGAN generator is trained using a noise vector and the category label as two inputs as shown in Figure 8a. The ACGAN discriminator has two outputs: one is the class probability value of the input image and the second is the category label of the input image. ACGAN is trained using a two-fold objective function which is given as:

$$L_G = E[\log P(S = \text{real}|X_{\text{real}})] + E\left[\log P\left(S = \text{Stego}|X_{\text{stego}}\right)\right]$$

$$L_C = E[\log P(C = C|X_{\text{real}})] + E[\log P(C = C|X_{\text{fake}})].$$

The generator is trained to maximize $L_C - L_G$ while Discriminator is trained to maximize $L_C + L_G$. $E(.)$ is the expectation value. $P(.)$ is the conditional probability.

The text information that needs to be secretly communicated is first encrypted using a code dictionary as shown in Figure 8b. The mapped label and noise Z are used as input to train the ACGAN generator for the generation of the stego image. The relationship between the labels and secret data is created using a mapping function. The trained ACGAN discriminator receives the stego image at the receiver end and produces a category label as output. The category label is decoded using the same code dictionary to extract the secret message.

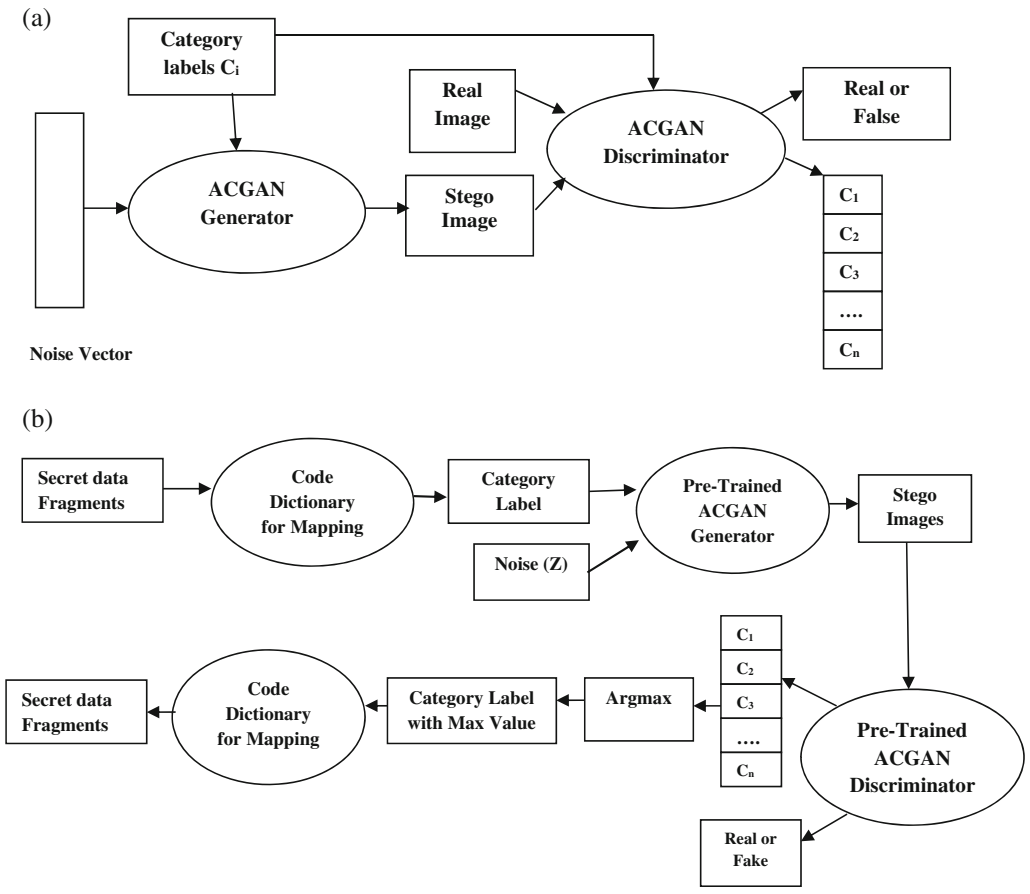

FIGURE 8 (a) ACGAN model. (b) Workflow of a noise-label steganography model

5.7 | Strong and weak features of deep learning based image steganography techniques

The strong and weak features of the main deep learning based steganography techniques are summarized below.

5.7.1 | Cover generation techniques

The advantage of generating cover images using GAN rather than using natural cover images is that the generated images are not known before and as such steganalyzers are not trained on these images and hence will be less susceptible to steganalysis. The disadvantage is that the cover generation approach is complex, and the generated cover images are not good enough and hence are suspicious to an intruder.

5.7.2 | Distortions learning techniques

Distortion learning techniques utilize GAN to automatically learn the distortion functions for steganography. No human-based rules are used for devising the distortion function. However, the security of such models is almost similar to the traditional state-of-art steganography algorithms.

5.7.3 | Adversarial image embedding techniques

Adversarial embedding methods hide the data in the adversarial examples to enhance the security of the stego image. However, the capacity of such methods is limited.

5.7.4 | GAN embedding techniques

GAN embedding techniques eliminate the need for human-based steganography algorithms and give the neural network full freedom to devise the algorithm on its own by adversarial training of the generator and steganalyzer networks. In addition to this, robust steganalyzers have been devised. The steganography images generated using this method outperform the traditional steganography methods in terms of security. The disadvantage is that such methods have limited embedding capacity and the quality of the generated images is poor (low imperceptibility or invisibility).

5.7.5 | Embedding less techniques

Embedding less techniques produce a stego image without embedding a secret message in a cover image which increases the security of steganography. However, the embedding capacity of this technique is low. In addition, all generated stego images are not real enough to fool the intruder

5.7.6 | Category label techniques

Category label techniques use conditional GAN for steganography where the conditional labels instead of secret data are used to produce a stego image. This improves security but the quality of generated stego images is poor. In addition, these techniques have limited capacity.

6 | PERFORMANCE COMPARISON OF IMAGE STEGANOGRAPHIC TECHNIQUES

The performance of image steganographic techniques is compared in this section. The results reported by researchers on benchmark data sets CelebA, Bossbase, PASCAL-VOC12, CIFAR-100, ImageNet, and USC-SIPI have been used to

evaluate the performance of various images steganography techniques. A summary of the datasets that have been used for comparing the performance is presented first.

6.1 | Datasets used for comparison

The commonly used datasets in image steganography are briefly summarized below.
(i) CelebA (CelebA Dataset, n.d.) is a large-scale dataset of celebrity faces. It consists of 202,599 face images of various celebrities across the globe with 10,177 distinctive identities. Each image has 40 attribute annotations and five landmark locations.
(ii) Bossbase (Bossbase Dataset, n.d.) is the benchmark dataset for steganography. It consists of 10,000 grayscale images of diverse scenes. The dataset has images of buildings, animals, landscapes, etc. Each image in the dataset is of size 512 * 512.
(iii) ImageNet (ImageNet Object Localization Challenge|Kaggle, n.d.) is a large-scale dataset of annotated images. It consists of approximately 14 million images of 21 thousand categories. Each category in the dataset consists of hundreds of images.
(iv) PASCAL-VOC12 (PASCAL VOC, 2012|*Kaggle*, n.d.) consists of images of 20 different object classes such as dogs, cats, sheep, persons, sofas, plants, etc. There is a bounding box annotation, object class annotation, and pixel-level segmentation annotation for all the images in the dataset.
(v) CIFAR-100 (CIFAR-10 and CIFAR-100 Datasets, n.d.) is a tiny color image dataset of 100 different object categories such as butterfly, mountain, lion, mouse, etc. It consists of 50,000 labeled images having 500 images of each class and 10,000 unlabeled images for testing. Each image is of size 32 * 32 * 3.
(vi) USC-SIPI (SIPI Image Database, n.d.) is a digital image database that consists of a collection of images such as brodatz and mosaic textures, high-altitude aerial images, mandrill, peppers, moving heads, moving vehicles, fly-overs, etc. The database is split into volumes having different image sizes of 256 * 256, 512 * 512, or 1024 * 1024. All images are 8 bits/pixel for black and white images, and 24 bits/pixel for color images.

6.2 | Security performance comparison

This section gives the security comparison of different steganography techniques. The security of a steganography technique is determined by the error rate of the steganalysis test performed by using a steganalyzer. The dataset used for the steganalysis test is split into two parts: the training dataset and the testing dataset. The steganalyzer is trained using the images in the training dataset and corresponding stego images. It is tested by using the testing dataset and the corresponding stego images. The error rate of the steganalysis test determines the security of a steganography technique. A higher error rate indicates better security and vice versa.

The security comparison of content-based steganography techniques such as WOW, HUGO, S-UNIWARD, and LSB is shown in Table 4. The steganalyzer employed in these methods is SRM except for (Lu et al., 2021) which uses RLCM-100D steganalyzer. All these methods employ the Bossbase dataset. The embedding capacity is set to 0.4 or 0.01 for the steganalysis test. From Table 4, it can be seen that content-based steganography methods have an error rate in the range of 10 to 25 which is not considered to be high.

The security performance of steganography approaches that use different cover generation techniques is performed. These techniques generate artificial cover images by using a given dataset. The cover generation models like SGAN and SSGAN are compared in Table 5, which shows the security performance of these cover generation techniques against the steganalysis test. These techniques use the CelebA dataset for training the model to generate artificial covers. The secret data is embedded in the generated covers using the LSB and HUGO methods with an embedding capacity of 0.4 bpp. The steganalyzer used by the SGAN is a self-defined steganalyzer whereas SSGAN employs a self-defined steganalyzer for LSB embedding and Qian's Net for HUGO embedding. The table shows that the SSGAN technique performs better than the SGAN technique for both HUGO and LSB embeddings. The results indicate that in general cover generation steganographic techniques are more secure than content-based steganographic techniques. In particular, the steganalysis test of the SSGAN technique has produced very good results with an error rate of more than 70%.

The security comparison of the steganography methods that automatically learn the distortion function for steganography using GAN is performed. The distortion learning steganography models like ASDL-GAN, UT-6HPF-GAN,

and UT-SCA-GAN are compared for security in Table 6. All these models employ the Bossbase dataset for the steganalysis test. The embedding capacity is set to 0.4 bpp. It employs SRM steganalyzer for the steganalysis. As depicted in the table, such techniques do not possess high resistance to steganalysis test and have security similar to that of the content-based traditional steganographic techniques.

The security performance of approaches that use the adversarial image embedding technique is performed. This technique uses adversarial images as cover images. The adversarial image embedding steganography models like ADV-EMD and CNN-Adv-EMD are compared in Table 7. The steganalysis test of the adversarial image embedding technique employs the Bossbase dataset. The embedding capacity is set to 0.4 bpp. The test makes use of XuNET steganalyzer. As depicted in the table, these models achieve higher error rates than the distortion learning steganography models, but lower error rates than cover generation steganography models. Thus adversarial image embedding steganography models are more secure than the distortion learning steganography models and less secure than cover generation steganography models.

The security performance comparison of the GAN embedding technique is performed. This technique utilizes deep learning models for the entire steganography process without using any rules framed by humans. The technique uses cover images from a given dataset. The security measure of GAN embedding steganographic models like HIGAN and the model described in Duan et al. (2019) is compared in Table 8. The steganalysis test uses the ImageNet dataset and XuNet steganalyzer. The model described in Duan et al. (2019) produces an appreciable error rate of 60.3% during the steganalysis test but this error rate is <72% obtained from the cover generation steganography techniques.

The security performance comparison of embedding less technique is performed. This technique directly generates stego images using noise and secret data without embedding the secret data in the cover image. The security comparison of embedding less steganography models like SsteGAN, GSS, and SWE-GAN is shown in Table 9. The steganalysis test of these models uses the CelebA dataset. As depicted in the table, the test of SsteGAN and GSS models produces an error rate of around 50%. The SWE-GAN model is tested for two cases: in case 1, stego images are kept private and are not used by the steganalyzer CRM for training. This results in low accuracy and high security; in case 2, stego images are made public and are used by the steganalyzer CRM for training. This decreases the error rate on the steganalyzer from 99.2% to 44% and hence security is decreased. These models have overall high security but when the stego data is made public there is an appreciable decline in the security.

The security performance comparison of category labels steganographic technique is performed. This technique directly generates stego images from the noise and the category labels. The category labels act as a driver for the stego image generation. The security performance comparison of the category labels models like CIH-GAN and SSS-GAN is shown in Table 10. CelebA dataset has been used for the steganalysis test. The CIH-GAN model has resulted in an error

TABLE 4 Security comparison of content-based traditional steganography techniques

References	Dataset	Embedding technique	Image size	Steganalyzer used	Relative capacity (bits per pixel)	Error of steganalyzer (%)
Pevný et al. (2010)	Bossbase	HUGO Algorithm	512 * 512	SRM	0.4	>10
Holub and Fridrich (2012)	Bossbase	WOW Algorithm	128 * 128	SRM	0.4	20
Holub et al. (2014)	Bossbase	S-UNIWARD Algorithm	512 * 512	SRM	0.4	20
Lu et al. (2021)	Bossbase	LSB with halftone images	256 * 256	RLCM-100D	0.01	25

TABLE 5 Security comparison of cover generation steganography techniques

Cover generation technique	Dataset	Relative capacity (bits per pixel)	Steganalyzer used	Embedding algorithm	Error of steganalyzer (%)
SGAN (Volkhonskiy et al., 2020)	CelebA	0.4	Self-defined	LSB	50
				HUGO	49.9
SSGAN (Shi et al., 2018)	CelebA	0.4	Self-defined	LSB	72
			Qian's Net	HUGO	71

rate of 52%. SSS-GAN has been tested for two cases: in case 1, stego images are kept hidden, and are not used by the steganalyzer CRM for training; in case 2, stego images are made public, and are used by the steganalyzer CRM for training. The error rate decreases from 99.9% to 56% when stego images are made public and hence decreases the security. These models have overall high security but when the stego data is made public the security decreases appreciably.

6.3 | Embedding capacity comparison

The embedding capacity performance comparison of the content-based steganographic technique is performed first. This technique makes use of rules framed by humans for the steganographic process. The content-based steganography models like OTPVD, EOTPVD, PBPVD, LSB + PVD, and LSB + QVD are compared in Table 11. The embedding capacity comparison test makes use of the USC-SIPI dataset with an image size of 512*512. The table shows that OTPVD, EOTPVD, PBPVD, and LSB + PVD models have embedding capacity in the range 2.483 to 4.55 bpb with PSNR in the range 33.06 to 40.4. LSB + QVD model has the highest embedding capacity of 4.55 bpb with a PSNR value of 33.06.

The embedding capacity comparison of the GAN embedding steganography technique is performed. The embedding capacity measure of GAN embedding steganography models like End-to-End CNN and the one described in Duan et al. (2019) is compared in Table 12. The models use the ImageNet dataset for the embedding capacity test. As depicted in the table, both the models have an embedding capacity of 8 bpp, but Duan et al. (2019) produces better PSNR value of 40.47.

6.4 | Invisibility comparison

The invisibility performance comparison of the content-based steganographic technique is performed first. Invisibility is determined using distortion measuring metrics such as PSNR or SSIM as discussed in Section 2. The embedding

TABLE 6 Security comparison of distortion learning steganography techniques

Method/references	Dataset	Relative capacity (bits per pixel)	Steganalyzer	Error of steganalyzer (%)
ASDL-GAN (Tang et al., 2017)	Bossbase	0.4	SRM	17
UT-SCA-GAN (Yang et al., 2018)	Bossbase	0.4	SRM	26
UT-6HPF-GAN (Yang et al., 2020)	Bossbase	0.4	SRM	29

TABLE 7 Security comparison of adversarial image embedding steganography techniques

Model/references	Dataset	Relative capacity (bits per pixel)	Embedding technique	Steganalyzer	Error of steganalyzer (%)
CNN-Adv-EMD (Tang et al., 2019)	Bossbase	0.4	Adversarial Embedding (STC)	XuNet	26.3
ADV-EMD (Zhang et al., 2018)	Bossbase	0.4	Adversarial Embedding (WOW)	XuNet	56.5

TABLE 8 Security comparison of GAN embedding steganography techniques

Cover-based techniques	Dataset	Steganalyzer used	Embedding technique	Relative capacity (bits per pixel)	Error of steganalyzer (%)
(Duan et al., 2019)	ImageNet	XuNet	Cover-Based	8	28.9
HIGAN (Fu et al., 2020)	ImageNet	XuNet	Cover-Based	8	60.3

capacity is set to 1 and 0.81 for this test. The invisibility performance of the difference expansion-based model and cubic reference table based model is compared in Table 13. Both the models use images of size 512 * 512 from the USC-SIPI dataset for the invisibility test. As depicted in the table, both models produce high PSNR values in the range of 43 to 50.

The invisibility comparison of the GAN embedding technique is performed. The distortion measure of GAN-embedding models like End-to-End CNN and ISGAN is compared in Table 14. These models use the PASCAL-VOC12 dataset for the distortion measure. The distortion is measured using PSNR and SSIM and embedding capacity is set to bpp. From the table, it is clear that these methods produce a PNSR value of 33 to 34 and an SSIM value in the range of 0.94 to 0.95.

From the above Tables, it can be concluded that the most secure content-based steganography techniques produce an error rate of 50% only for the steganalysis test (Table 5). Embedding capacity and image quality of the steganography process is not very high. Fully deep learning techniques and hybrid steganography techniques have been explored by researchers to improve the performance of steganography. Embedding data in artificial covers or adversarial images has resulted in better security than the traditional methods. However, distortion learning methods proposed so far are not more secure than traditional distortion minimization methods (Table 6). Fully deep learning techniques such as GAN embedding methods have achieved very good results in terms of security and capacity compared to the traditional and hybrid methods. Embedding Capacity has been increased (Table 12). SWE-GAN and SSS-GAN have achieved high error rate on different steganalyzers when stego images are kept hidden (Tables 9 and 10). However, the error rate decreases appreciably to around when the stego images are made public. Further, the embedding capacity of these methods is limited. The invisibility of the deep learning methods measured by PSNR and SSIM is good but not better than traditional methods. A lot of work has been done in deep learning based steganography in terms of

TABLE 9 Security comparison of embedding less steganography technique

Generative models	Dataset	Relative capacity (bits per pixel)	Embedding technique	Steganalyzer used	Error of steganalyzer (%)
SsteGAN (Wang et al., 2018)	CelebA	0.4	Noise-based	Self-defined	40.84
GSS (Zhang, Fu, et al., 2019)	CelebA	0.4	Noise-based	SPAM	<50
SWE-GAN (Hu et al., 2018) Case1	CelebA	0.0732	Noise-based	CRM	99.2
SWE-GAN (Hu et al., 2018) *Case 2*				*CRM*	*44*

TABLE 10 Security comparison of category label techniques

Generative models	Dataset	Embedding technique	Steganalyzer used	Relative capacity (bits per pixel)	Error of steganalyzer (%)
CIH-GAN (Liu, Zhou, et al., 2018)	MNIST	Noise-label	EC + SPAM	$9.125 * 10^{-5}$	52
SSS-GAN (Zhang et al., 2020) Case1	MNIST	Noise-label	CRM	$9.125 * 10^{-5}$	99.9
SSS-GAN (Zhang et al., 2020) Case 2			CRM		56

TABLE 11 Embedding capacity comparison of traditional steganography techniques

References	Dataset	Embedding technique	Image size	Capacity (bits per byte)	Average PSNR
Grajeda-Marín et al. (2018)	USC-SIPI	OTPVD	512 * 512	2.483	37.774
		EOTPVD		2.733	37.635
Hussain et al. (2017)	USC-SIPI	PBPVD + iRMDR	512 * 512	3	38.84
Swain (2016)	USC-SIPI	LSB + PVD	512 * 512	3.1	40.4
Swain (2019)	USC-SIPI	LSB + QVD	512 * 512	4.55	33.06

TABLE 12 Embedding capacity comparison of GAN embedding steganography models

Models	Embedding technique	Dataset	Relative capacity (bits per pixel)	Average PSNR
End-to-End CNN(ur Rehman et al., 2019)	Cover-Based	ImageNet	8	32.9
(Duan et al., 2019)	Cover-Based	ImageNet	8	40.47

TABLE 13 Invisibility comparison of traditional steganography techniques

References	Dataset	Embedding technique	Image size	Average PSNR	Relative capacity (bpp)
Chang et al. (2017)	USC-SIPI	Difference Expansion	512 * 512	43	0.81
Liao et al. (2018)	USC-SIPI	CRT	512 * 512	48.615	1
		CRT-PVD		49.41	1

TABLE 14 Distortion measurement of GAN embedding techniques for improved imperceptibility

			Stego-cover (encoding)		Recovered-secret (decoding)		
Model/references	Cover	Stego (secret)	Average PSNR (db)	Average SSIM (db)	Average PSNR (db)	Average SSIM (db)	Relative capacity (bpp)
End-to-End CNN (ur Rehman et al., 2019)	PASCAL-VOC12	PASCAL-VOC12	33.7	0.96	35.9	0.95	8
ISGAN (Zhang, Fu, et al., 2019)	PASCAL-VOC12	PASCAL-VOC12	34.49	0.9661	33.31	0.9467	8

security. There is a scope for improving the deep learning steganographic techniques that can improve the capacity and invisibility while keeping the security high. There is also a need for new suitable deep learning architectures that can improve the capacity and invisibility of image steganography.

7 | CONCLUSION

A review of deep learning based image steganography techniques was carried out in this paper. The techniques were divided into various categories and sub-categories that were based on the embedding strategy employed. The three criteria, namely security, capacity, and invisibility for measuring the performance of steganography were described, and were used to compare the performance of various deep learning based steganography techniques. The benchmark data sets CelebA, Bossbase, PASCAL-VOC12, CIFAR-100, and ImageNet were used for evaluating the performance of various image steganography techniques. It was observed from the results that the capacity of deep learning based models is still less than the traditional techniques. Future work can include developing an improved deep learning based image steganography technique that has better embedding capacity. There is also a scope to improve the imperceptibility of the deep learning based image steganography technique as little work has been done on improving imperceptibility.

AUTHOR CONTRIBUTIONS

Mohd Arif Wani: Writing – original draft (lead). **Bisma Sultan:** Writing – original draft (equal).

DATA AVAILABILITY STATEMENT

NA

ORCID

Mohd Arif Wani https://orcid.org/0000-0002-4178-3588

REFERENCES

Abadi, M., & Andersen, D. G. (2016). Learning to protect communications with adversarial neural cryptography. *arXiv*, 1–15. http://arxiv.org/abs/1610.06918

Ahuja, S., Kumar, C. U., & Hemalatha, S. (2019). Competitive coevolution for color image steganography. 2019 International conference on intelligent computing and control systems, ICCS 2019, ICICCS, 719–723. https://doi.org/10.1109/ICCS45141.2019.9065844

Baluja, S. (2017). Hiding images in plain sight: Deep steganography. In *Advances in neural information processing systems, 2017-Decem(nips)* (pp. 2070–2080). Curran Associates, Inc.

Bossbase Dataset. (n.d.). Retrieved from https://dde.binghamton.edu/download/

CelebA Dataset. (n.d.). Retrieved from https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Chang, C. C., Huang, Y. H., & Lu, T. C. (2017). A difference expansion based reversible information hiding scheme with high stego image visual quality. *Multimedia Tools and Applications*, *76*(10), 12659–12681. https://doi.org/10.1007/s11042-016-3689-3

CIFAR-10 and CIFAR-100 datasets. (n.d.). Retrieved from https://www.cs.toronto.edu/kriz/cifar.html

Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., & Qin, C. (2019). Reversible image steganography scheme based on a U-net structure. *IEEE Access*, *7*, 9314–9323. https://doi.org/10.1109/ACCESS.2019.2891247

Fridrich, J., & Kodovsky, J. (2012). Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, *7*(3), 868–882. https://doi.org/10.1109/TIFS.2012.2190402

Fu, Z., Wang, F., & Cheng, X. (2020). The secure steganography for hiding images via GAN. *EURASIP Journal on Image and Video Processing*. https://doi.org/10.1186/s13640-020-00534-2

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *3rd international conference on learning representations, ICLR 2015: conference track proceedings* (pp. 1–11). https://arxiv.org/abs/1412.6572

Grajeda-Marín, I. R., Montes-Venegas, H. A., Marcial-Romero, J. R., Hernández-Servín, J. A., Muñoz-Jiménez, V., & Luna, G. D. I. (2018). A new optimization strategy for solving the fall-off boundary value problem in pixel-value differencing steganography. *International Journal of Pattern Recognition and Artificial Intelligence*, *32*(1), 1–17. https://doi.org/10.1142/S0218001418600108

Hayes, J., & Danezis, G. (2017). Generating steganographic images via adversarial training. In *Advances in neural information processing systems, 2017-Decem* (pp. 1955–1964). Curran Associates Inc.

Holub, H., & Fridrich, J. (2012). Designing steganographic distortion using directional filters. *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, 234–239. https://doi.org/10.1109/WIFS.2012.6412655

Holub, V., Fridrich, J., & Denemark, T. (2014). Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, *2014*, 1–13. https://doi.org/10.1186/1687-417X-2014-1

Hu, D., Wang, L., Jiang, W., Zheng, S., & Li, B. (2018). A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, *6*, 38303–38314. https://doi.org/10.1109/ACCESS.2018.2852771

Huang, J., Cheng, S., Lou, S., & Jiang, F. (2019). Image steganography using texture features and GANs. Proceedings of the International Joint Conference on Neural Networks, 2019-July(July), 1–8. https://doi.org/10.1109/IJCNN.2019.8852252

Hussain, M., Abdul Wahab, A. W., Ho, A. T. S., Javed, N., & Jung, K. H. (2017). A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. *Signal Processing: Image Communication*, *50*, 44–57. https://doi.org/10.1016/j.image.2016.10.005

ImageNet Object Localization Challenge|Kaggle. (n.d.). Retrieved from https://www.kaggle.com/c/imagenet-object-localization-challenge/data?select=imagenet_object_localization_patched2019.tar.gz

Ke, Y., Zhang, M., Liu, J., Su, T., & Yang, X. (2017). *Generative Steganography with Kerckhoffs' Principle based on Generative Adversarial Networks*. 1–5. http://arxiv.org/abs/1711.04916

Kumar, V., & Kumar, D. (2018). A modified DWT-based image steganography technique. *Multimedia Tools and Applications*, *77*(11), 13279–13308. https://doi.org/10.1007/s11042-017-4947-8

Lerch-Hostalot, D., & Megías, D. (2016). Unsupervised steganalysis based on artificial training sets. *Engineering Applications of Artificial Intelligence*, *50*(April), 45–59. https://doi.org/10.1016/j.engappai.2015.12.013

Li, C., Jiang, Y., & Cheslyar, M. (2018). Embedding image through generated intermediate medium using deep convolutional generative adversarial network. *Computers, Materials and Continua*, *56*(2), 313–324. https://doi.org/10.3970/CMC.2018.03950

Li, N., Hu, J., Sun, R., Wang, S., & Luo, Z. (2017). A high-capacity 3D steganography algorithm with adjustable distortion. *IEEE Access*, *5*, 24457–24466. https://doi.org/10.1109/ACCESS.2017.2767072

Li, S., & Zhang, X. (2019). Toward construction-based data hiding: From secrets to fingerprint images. *IEEE Transactions on Image Processing*, *28*(3), 1482–1497. https://doi.org/10.1109/TIP.2018.2878290

Liao, X., Guo, S., Yin, J., Wang, H., Li, X., & Sangaiah, A. K. (2018). New cubic reference table based image steganography. *Multimedia Tools and Applications*, *77*(8), 10033–10050. https://doi.org/10.1007/s11042-017-4946-9

Liao, X., Yin, J., Chen, M., & Qin, Z. (2022). Adaptive payload distribution in multiple images steganography based on image texture features. *IEEE Transactions on Dependable and Secure Computing*, *19*(2), 897–911. https://doi.org/10.1109/TDSC.2020.3004708

Liao, X., Yu, Y., Li, B., Li, Z., & Qin, Z. (2020). A new payload partition strategy in color image steganography. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(3), 685–696. https://doi.org/10.1109/TCSVT.2019.2896270

Liu, J., Zhou, T., Zhang, Z., Ke, Y., Lei, Y., & Zhang, M. (2018). Digital cardan grille: A modern approach for information hiding. *ACM International Conference Proceeding Series*, 441–446. https://doi.org/10.1145/3297156.3297255

Liu, M. M., Zhang, M. Q., Liu, J., Gao, P. X., & Zhang, Y. N. (2018). Coverless information hiding based on generative adversarial networks. *Yingyong Kexue Xuebao/Journal of Applied Sciences*, *36*(2), 371–382. https://doi.org/10.3969/j.issn.0255-8297.2018.02.015

Lu, T. C., Tseng, C. Y., & Wu, J. H. (2015). Dual imaging-based reversible hiding technique using LSB matching. *Signal Processing*, *108*, 77–89. https://doi.org/10.1016/j.sigpro.2014.08.022

Lu, W., Xue, Y., Yeung, Y., Liu, H., Huang, J., & Shi, Y. Q. (2021). Secure halftone image steganography based on pixel density transition. *IEEE Transactions on Dependable and Secure Computing*, *18*(3), 1137–1149. https://doi.org/10.1109/TDSC.2019.2933621

Luo, T., Jiang, G., Yu, M., Xu, H., & Gao, W. (2018). Sparse recovery based reversible data hiding method using the human visual system. *Multimedia Tools and Applications*, *77*(15), 19027–19050. https://doi.org/10.1007/s11042-017-5356-8

Ma, S., Zhao, X., & Liu, Y. (2019). Adaptive spatial steganography based on adversarial examples. *Multimedia Tools and Applications*, *78*(22), 32503–32522. https://doi.org/10.1007/s11042-019-07994-3

Mandal, J. K., Satapathy, S. C., Sanyal, M. K., & Bhateja, V. (2017). Preface. *Advances in Intelligent Systems and Computing*, *458*, V–VI. https://doi.org/10.1007/978-981-10-2035-3

Mielikainen, J. (2006). LSB matching revisited. *IEEE Signal Processing Letters*, *13*(5), 285–287. https://doi.org/10.1109/LSP.2006.870357

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

Navab, N., Hornegger, J., Wells, W. M., & Frangi, A. F. (2015). Medical image computing and computer-assisted intervention: MICCAI 2015: 18th international conference Munich, Germany, October 5–9, 2015 proceedings, part III. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, *9351*(Cvd), 12–20. https://doi.org/10.1007/978-3-319-24574-4

Nyeem, H. (2018). Reversible data hiding with image bit-plane slicing. 20th international conference of computer and information technology, ICCIT 2017, 2018-Janua(December), 1–6. https://doi.org/10.1109/ICCITECHN.2017.8281763

PASCAL VOC 2012 *Kaggle*. (n.d.). Retrieved from https://www.kaggle.com/huanghanchina/pascal-voc-2012

Pevný, T., Filler, T., & Bas, P. (2010). Using high-dimensional image models to perform highly undetectable steganography. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, *6387 LNCS*, 161–177. https://doi.org/10.1007/978-3-642-16435-4_13

Qian, Y., Dong, J., & Wang, W. (2015). Deep learning for steganalysis via convolutional neural networks. *Proceedings of SPIE - The International Society for Optical Engineering*, 9409, 94 090J–94 090J–10. https://doi.org/10.1117/12.2083479

Qu, Z., Cheng, Z., Liu, W., & Wang, X. (2019). A novel quantum image steganography algorithm based on exploiting modification direction. *Multimedia Tools and Applications*, *78*(7), 7981–8001. https://doi.org/10.1007/s11042-018-6476-5

Rabie, T., & Kamel, I. (2017). High-capacity steganography: A global-adaptive-region discrete cosine transform approach. *Multimedia Tools and Applications*, *76*(5), 6473–6493. https://doi.org/10.1007/s11042-016-3301-x

Rajendran, S., & Doraipandian, M. (2017). Chaotic map based random image steganography using LSB technique. *International Journal of Network Security*, *19*(4), 593–598. https://doi.org/10.6633/IJNS.201707.19(4).12

Shi, H., Dong, J., Wang, W., Qian, Y., & Zhang, X. (2018). SSGAN: Secure steganography based on generative adversarial networks. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 10735). Springer International Publishing. https://doi.org/10.1007/978-3-319-77380-3_51

Shi, H., Zhang, X. Y., Wang, S., Fu, G., & Tang, J. (2019). Synchronized detection and recovery of Steganographic messages with adversarial learning. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11537, pp. 31–43). Springer Nature. https://doi.org/10.1007/978-3-030-22741-8_3

SIPI Image Database (n.d.). Retrieved from https://sipi.usc.edu/database/

Sultan, B., & Wani, M. A. (2021). Generative adversarial network based steganography with different color spaces. Proceedings of the 2021 8th international conference on computing for sustainable global development, INDIACom 2021, 112–119. https://doi.org/10.1109/INDIACom51348.2021.00021

Swain, G. (2016). A steganographic method combining LSB substitution and PVD in a block. *Procedia Computer Science*, *85*, 39–44. https://doi.org/10.1016/J.PROCS.2016.05.174

Swain, G. (2019). Very high capacity image steganography technique using quotient value differencing and LSB substitution. *Arabian Journal for Science and Engineering*, *44*(4), 2995–3004. https://doi.org/10.1007/s13369-018-3372-2

Swain, G., & Lenka, S. K. (2015). Pixel value differencing steganography using correlation of target pixel with neighboring pixels. Proceedings of 2015 IEEE international conference on electrical, computer and communication technologies, ICECCT 2015. https://doi.org/10.1109/ICECCT.2015.7226029

Tan, J., Liao, X., Liu, J., Cao, Y., & Jiang, H. (2022). Channel attention image steganography With generative adversarial networks. *IEEE Transactions on Network Science and Engineering*, *9*(2), 888–903. https://doi.org/10.1109/TNSE.2021.3139671

Tang, W., Li, B., Tan, S., Barni, M., & Huang, J. (2019). CNN-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, *14*(8), 2074–2087. https://doi.org/10.1109/TIFS.2019.2891237

Tang, W., Tan, S., Li, B., & Huang, J. (2017). Automatic Steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, *24*(10), 1547–1551. https://doi.org/10.1109/LSP.2017.2745572

ur Rehman, A., Rahim, R., Nadeem, S., & ul Hussain, S. (2019). End-to-end trained CNN encoder-decoder networks for image steganography. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 11132 LNCS, 723–729. https://doi.org/10.1007/978-3-030-11018-5_64

Volkhonskiy, D., Nazarov, I., & Burnaev, E. (2020). Steganographic generative adversarial networks. 97. https://doi.org/10.1117/12.2559429

Wang, Z., Gao, N., Wang, X., Qu, X., & Li, L. (2018). SSteGAN: Self-learning steganography based on generative adversarial networks. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11302). Springer International Publishing. https://doi.org/10.1007/978-3-030-04179-3_22

With, T. (2019). Integrated steganography and steganalysis. 1–14.

Wu, P., Yang, Y., & Li, X. (2018a). Image-into-image steganography using deep convolutional network. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11165, pp. 792–802). Springer Nature. https://doi.org/10.1007/978-3-030-00767-6_73

Wu, P., Yang, Y., & Li, X. (2018b). StegNet: Mega image steganography capacity with deep convolutional network. *Future Internet*, *10*(6), 1–15. https://doi.org/10.3390/FI10060054

Xu, G., Wu, H. Z., & Shi, Y. Q. (2016). Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, *23*(5), 708–712. https://doi.org/10.1109/LSP.2016.2548421

Yang, J., Liu, K., Kang, X., Wong, E. K., & Shi, Y.-Q. (2018). Spatial image steganography based on generative adversarial network. *arXiv*, *1*, 1–7. http://arxiv.org/abs/1804.07939

Yang, J., Ruan, D., Huang, J., Kang, X., & Shi, Y. Q. (2020). An embedding cost learning framework using GAN. *IEEE Transactions on Information Forensics and Security*, *15*, 839–851. https://doi.org/10.1109/TIFS.2019.2922229

Yang, K., Chen, K., Zhang, W., & Yu, N. (2019). Provably secure generative steganography based on autoregressive model. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11378). Springer International Publishing. https://doi.org/10.1007/978-3-030-11389-6_5

Ye, J., Ni, J., & Yi, Y. (2017). Deep learning hierarchical representations for image Steganalysis. *IEEE Transactions on Information Forensics and Security*, *12*(11), 2545–2557. https://doi.org/10.1109/TIFS.2017.2710946

Yedroudj, M., Chaumont, M., & Comby, F. (2018). How to augment a small learning set for improving the performances of a CNN-based steganalyzer? https://doi.org/10.48550/arxiv.1801.04076

Yedroudj, M., Comby, F., & Chaumont, M. (2018). Yedroudj-net: An efficient CNN for spatial Steganalysis. In *ICASSP, IEEE international conference on acoustics, speech and signal processing - proceedings, 2018-April(1)* (pp. 2092–2096). https://doi.org/10.1109/ICASSP.2018.8461438

Yedroudj, M., Comby, F., & Chaumont, M. (2020). Steganography using a 3-player game. *Journal of Visual Communication and Image Representation*, *72*, 102910. IEEE. https://doi.org/10.1016/j.jvcir.2020.102910

Zhang, K. A., Cuesta-Infante, A., Xu, L., & Veeramachaneni, K. (2019). *SteganoGAN: High capacity image steganography with GANs*. http://arxiv.org/abs/1901.03892

Zhang, R., Dong, S., & Liu, J. (2019). Invisible steganography via generative adversarial networks. *Multimedia Tools and Applications*, *78*(7), 8559–8575. https://doi.org/10.1007/s11042-018-6951-z

Zhang, W., Wang, H., Hou, D., & Yu, N. (2016). Reversible data hiding in encrypted images by reversible image transformation. *IEEE Transactions on Multimedia*, *18*(8), 1469–1479. https://doi.org/10.1109/TMM.2016.2569497

Zhang, Y., Zhang, W., Chen, K., Liu, J., Liu, Y., & Yu, N. (2018). Adversarial examples against deep neural network based steganalysis. In proceedings of the 6th ACM workshop on information hiding and multimedia security (IH&MMSec '18) (pp. 67–72). Association for Computing Machinery. https://doi.org/10.1145/3206004.3206012

Zhang, Z., Fu, G., Di, F., Li, C., & Liu, J. (2019). Generative reversible data hiding by image-to-image translation via GANs. *Security and Communication Networks*, *2019*, 1–10. https://doi.org/10.1155/2019/4932782

Zhang, Z., Fu, G., Ni, R., Liu, J., & Yang, X. (2020). A generative method for steganography by cover synthesis with auxiliary semantics. *Tsinghua Science and Technology*, *25*(4), 516–527. https://doi.org/10.26599/TST.2019.9010027

Zhang, Z., Liu, J., Ke, Y., Lei, Y., Li, J., Zhang, M., & Yang, X. (2019). Generative steganography by sampling. *IEEE Access*, *7*, 118586–118597. https://doi.org/10.1109/ACCESS.2019.2920313

Zhou, L., Feng, G., Shen, L., & Zhang, X. (2020). On security enhancement of steganography via generative adversarial image. *IEEE Signal Processing Letters*, *27*, 166–170. https://doi.org/10.1109/LSP.2019.2963180

Zhu, J., Kaplan, R., Johnson, J., & Fei-Fei, L. (2018). HiDDeN: Hiding data with deep networks. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11219). Springer International Publishing. https://doi.org/10.1007/978-3-030-01267-0_40

Zi, H., Zhang, Q., Yang, J., & Kang, X. (2019). Steganography with convincing Normal image from a joint generative adversarial framework. 2018 Asia-Pacific signal and information processing association annual summit and conference, APSIPA ASC 2018: proceedings, November, 526–532. https://doi.org/10.23919/APSIPA.2018.8659716

How to cite this article: Wani, M. A., & Sultan, B. (2023). Deep learning based image steganography: A review. *WIREs Data Mining and Knowledge Discovery*, *13*(3), e1481. https://doi.org/10.1002/widm.1481