
# Survey of Research Opportunities that use Artificial Intelligence in Image Steganography

Ayyah Abdulhafidh Mahmoud Fadhl[1], Bander Ali Saleh Al-rimy[2], Sultan Ahmed Almalki[3,*], Tami Abdulrahman Alghamdi[4], Azan Hamad Alkhorem[5], Frederick T. Sheldon[6]

[1]Artificial Intelligence Department, Libyan International University, Benghazi, Libya
[2]School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK
[3]Computer Department, Applied College, Najran University, Najran 66462 , Kingdom of Saudi Arabia
[4]Computer Science Department, Faculty of Computing and Information, Al-Baha University, Al-Baha, 65779, Saudi Arabia
[5]Department of Computer Engineering, College of Computer Science and Information Technology, Majmaah University, Majmaah, 11952, Saudi Arabia
[6]Department of Computer Science, University of Idaho, Moscow, ID 83844, USA

Emails: ayyah.fadhi@limu.edu.ly; bander.al-rimy@port.ac.uk; Saalmalki@nu.edu.sa; Talwajeeh@bu.edu.sa; ah.alkhorem@mu.edu.sa; sheldon@uidaho.edu

**Abstract**

Steganography conceals ”secrets” within an convenient and expedient multimedia *carrier*. The carrier could be text (i.e., not plain text), images, audio and/or video files (i.e., carrier channels). The fact that concealed information is contained in the otherwise ordinary and mundane carrier file is known only by the sender-receiver pair. Only they share the existence of the secret. Images are the most popular (i.e., multimedia) carriers because of their inherent property that enables better obfuscation. Content adaptive image steganography is a new trend in the field for messaging secrets inside unsuspected image file transfers. As the name suggests, the embedding locations are altered adaptively depending on the image content that optimizes the decision of choosing a location inside the carrier so that an embedding is not discernible (i.e., additive distortion is minimized). Herein, we critique the various approaches used for content-adaptive image steganography which can be broadly categorized as CNN-based, GAN-based, along with minimizing additive distortion function-based.
We provide a brief historical account toward better anticipating the future research opportunities in terms of properties, and evaluation metrics. A summary table of these past and future directions is provided. Moreover, we highlight trends along with their concomitant advantages and disadvantages toward identifying opportunity gaps.

**Keywords:** Content Adaptive Image Steganography; Deep learning-based steganography; Steganalysis; Additive distortion

## 1 Introduction

Three essential properties of steganography: i) security, ii) capacity, and iii) robustness, create a useful method including text, image, video, or audio content for information transfer via covert communication channels. Technological advances, particularly digital data communication, have contributed to the widening usage of data transmission. Now, data exchange is a phenomenon of everyday life.[35] An obvious example is the exponential growth

*DOI: https://doi.org/10.54216/FPA.180216*
*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

of social networks, photo-bound commercial websites, and unrelenting email messages/attachments, and streaming video. As a result, keeping data secure protected from malicious threat actors, eavesdropping, and other nefarious activities jeopardize confidentiality, integrity and availability (CIA). Cryptography, is a baseline prominent data security branch and the most common means of ensuring CIA.

In this popular and well exercised scenario, different data encryption techniques are applied to conceal plain text data into cipher text on the sender side using an encryption key, transmitting the cipher text, and then converting the data back from the cipher text at the receiver side using a decryption key to plain text. [We should add a citation to an article here that covers the recent history of cryptography adding that different forms of cryptography have been in existence almost as long as man has walked the earth, B.C. ]

Cryptography maintains data security, or more precisely CIA. While steganography, by making the content of the secret message or original data *invisible* to unauthorized parties, or cryptanalysis, the message existence is not denied while simultaneously the cipher text is still visible .[35] Consequently, information hiding, specifically steganography, which is another data security technique, is used to maintain data confidentiality by making message existence even more secure by denying its existence.[19]

Thus, steganography is the science of writing concealed messages in such a way that no one but the intended recipient is aware of their presence. Digital steganography conceals secret data inside a digital medium, such as digital images, audio, text, or video files. Despite that cryptography ensures data integrity and non-repudiation in addition to confidentiality and authentication, which are also cooperatively maintained by steganography.

What distinguishes steganography is that the data structure used to hold the secret message is typically not altered, which is more suitable for certain kinds of messaging, such as the enormous amount of data being sent over the Internet and social media platforms. This emerging trend has contributed to its growing popularity over the past few years. The choice of images as the carrier, or image steganography as the messaging protocol has become a primary research focus due to the convenience of internet-based multimedia communications.[15]

Adaptive image steganography, a promising new trend in the steganographic field .[19] Content-adaptive image steganography basically means that the embedding locations are altered adaptively depending on the image content, such as its texture and smooth regions.[19]

Covert communications or digital data transmission is not the only steganography application .[7] Other applications, such as copyright control of materials, smart identity cards with embedded personal information, video error correction during transmission ,[22,30] privacy protection of authorized individuals in surveillance systems ,[44] and TCP/IP packets (for example, a unique ID can be embedded in an image to analyze the network traffic of specific users) .[18]

/todoNon-fungible tokens, often referred to as NFTs, are blockchain-based tokens that each represent a unique asset like a piece of art, digital content, or media. An NFT can be thought of as an irrevocable digital certificate of ownership and authenticity for a given asset, whether digital or physical.

Recent years have seen numerous publications on this subject. Accordingly, the primary objective of this paper is to review and analyze several content-adaptive image steganography approaches. But before that, a general overview of the topic is given, along with evaluation metrics.

The remainder of the paper is structured as follows. Section II provides an overview of image steganography. Recent research is summarized in Section III in terms of three approaches: CNN-Based, GAN-Based, and Minimizing Additive Distortion Function-Based. Section IV compares, and discusses the advantages and disadvantages of each approach. Section V concludes with a brief summary.

## 2 Image Steganography Overview

### 2.1 Steganography Evolution and Background

The name ”Steganography” results from the combination of two Greek words, ”Stegano”, which means ”cover or secret,” and ”Graphy,” which means ”writing or drawing”. The first employment of steganography, or its

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

early version, was around 440 B.C., as stated by the Greek historian Herodotus. Histaiacus, the Greek ruler, was imprisoned in the 5th century B.C. by King Darius in Susa. During that time, he wanted to communicate with his son-in-law, Aristagoras, in Miletus secretly. Therefore, he shaved his trusted slave, tattooed a secret message on his scalp, and sent the slave to deliver the message after ensuring that the message got hidden by the growing hair .[15,19] In this early version of steganography, the safety of the message was guaranteed. Nevertheless, those were not the only applications of steganography by the Greeks in antiquity. In fact, the region was famous for passing via secret messaging. Using a wax-layered tablet, Demerstus sent a message to the Spartans, warning them of Xerxes's eminent invasion. Demerstus scraped the wax, scratched his messages on the wooden tablet, and then layered the tablet with wax again to hide the scratched message .[19,34] The Chinese also made their own contribution to the long-established pedigree of steganography. Their contribution demonstrates the use of a shared mask _grid holed paper_to write a secret message before filling the blank paper to make it appear innocent .[19] Invisible ink is another early version of steganography which relies on some probabilities of chemical reactions. Invisible ink was first used by ancient Greeks and Romans more than 2,000 years ago using readily available substances such as urine, milk, and lemon juice to write between the lines of text, and then, at the receiver side, another concoction was used to make the invisible writing visible.

During both World Wars I and II, the utilization of steganography for sending messages secretly was common place. Several steganographic methods were prevalent, including Microdot Technology. A Microdot, which is a photograph reduced to the period or dot size, had been developed by Professor Zapp and was being used by the Germans. The Zapp invention reduced photographically secret messages and affixed them to any dot or punctuation in a cover text. The FBI director at the time, Edgar Hoover, gave it the name Microdot after discovering it in a typed envelop carried by a German agent in 1941. Moreover, a modified version of invisible inks were used by the British, Americans, and Germans whom mastered their use during both world wars .[19] Other stenographic techniques used include open coded messages, different null ciphers, and the enigma machines [27]. These various methods of sending/receiving encoded secrete messaged play an exceptionally important role.

As a result of the third industrial revolution, or as it is known by the term "digital revolution," which occurred in the second half of the 20th century, powered by the invention of the semi-conductor, the invention of the personal computer and the internet have emerged [4]. Inevitably, evolution in numerous fields and the movement towards digitization has resulted. The primary two factors that gave birth to digital steganography were the growth in computer processing power combined with Internet speed and throughput [26]. Digital steganography can be defined in short as a secure data transmission technique. It was first studied by cryptographer Gustavus Simmons in 1983 .[10] He portrayed steganography by using the prisoner's problem. This problem scenario includes two inmates, Alice and Bob, who are aiming to hatch an escape plan together. However, the only communication channel available for them is insecure by virtue of the systematic monitoring by the prison's warden Eva shown in Figure 1. Eva will throw Alice and Bob into solitary confinement if she suspects any kind of scrambled communication. Therefore, both Alice and Bob need to find a way to conceal their messages by making them "invisible." The prisoner's problem not only illustrates the steganography process and reason, it also highlights the steganalysis role illustrated by Eva. In short, steganalysis is used to detect the existence of hidden information. Therefore, it is worth saying steganography and steganalysis are two adversarial roles. Beginning from their discovery, they have been influenced by each other's and are characterized appropriately as min-max games .[15] A more contemporary definition of digital steganography is to be next discussed.

With the constant improvement of Industrial Revolution 4 (IR4) and the development of the Convolutional Neural Network (CNN), which was first brought to light by the work of LeCuN et. al [20] in 1989, both steganography and steganalysis started moving in this direction. The first application of CNN in steganalysis was conducted in 2014 by Shunquan Tan and Bin Li, when they proposed a nine-layer, three-stage CNN called Stack Convolutional Auto Encoder (SCAE) [36] based blind steganalysis that aimed to outperform the well-known Spatial Domain Rich Model (SRM) [9] based steganalysis. However, their outcomes were not satisfactory for different purposes, but that was a good start at that time. In the same manner, CNN has been utilized in steganography as well. In 2014, GAN, a new paradigm of machine learning algorithms, was designed by Lan Goodfellow and his colleagues.[11] GANs perfectly fit the steganography pattern by simulating a minmax game. Therefore, much work has utilized the GAN paradigm to establish different steganography architectures.

### 2.2 Digital Image Steganography Definition.

A block diagram representing digital image steganography is shown in figure 1. On the senders side, the digital image used to hide the secret image is known as the cover image. The input for any steganographic system is a

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

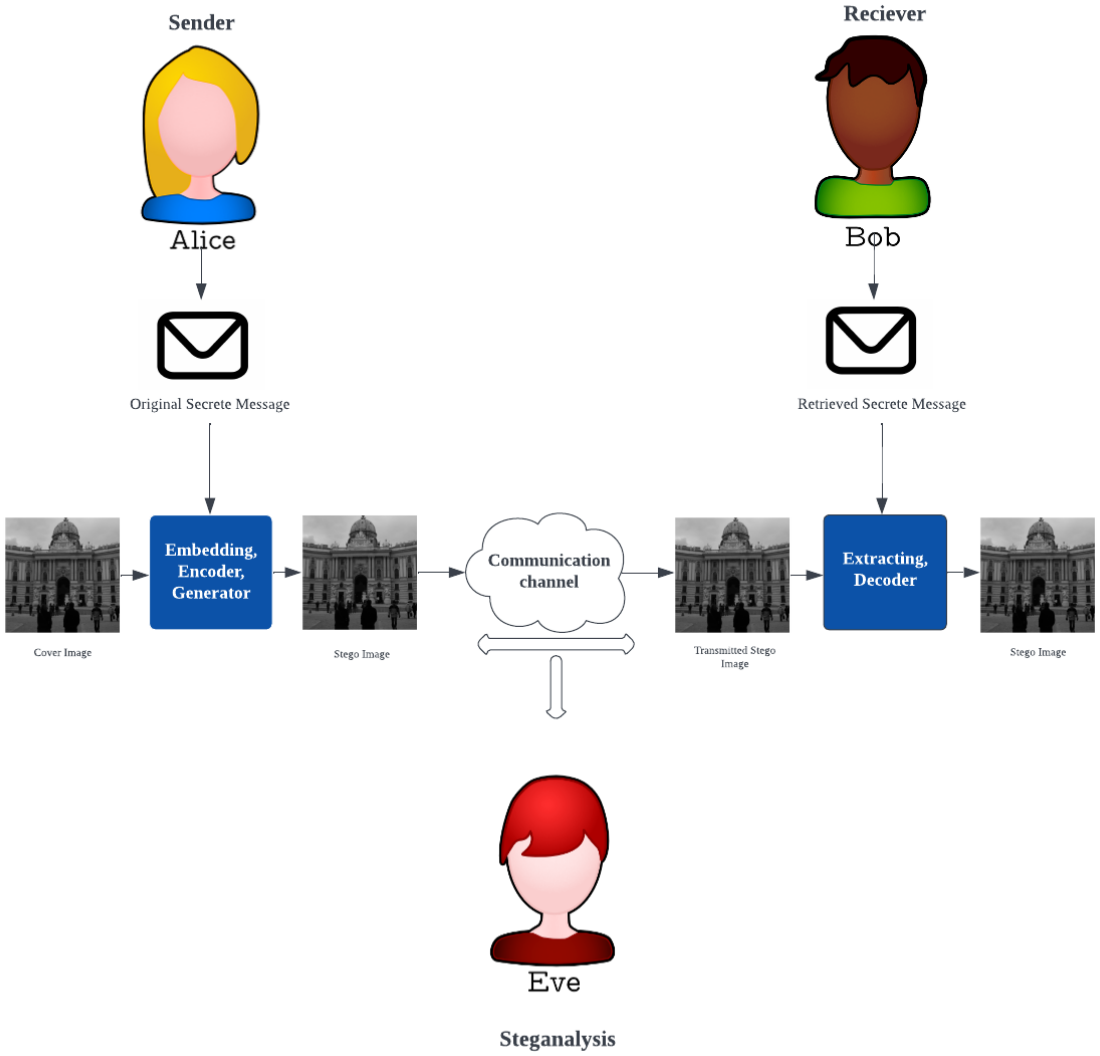


Figure 1: Steganography definition using the Prisoner problem. Alice or the sender embeds a secret message into a cover image before transmitting through insecure channel. Eva, the steganalysis actor, tries to detect the existence of a secret message. Bob extracts the secret message from the steganographic image on the receiver side.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

cover image and a secret message, and the output of the same system is a "Stego" image. The whole process is expressed by the embedding function. Refer to Eq. 1, where C is the cover image, S is the secret message, and $\bar{C}$ is the stego image.

$$\overline{C} = Em(C, S), \tag{1}$$

When the Stego image is output from the embedding function, it arrives at the receiver side after travelling through an insecure channel and subject to steganalysis. At the receiver side, the opposite process is performed; the secret message is extracted from the cover image with the help of an extracting schema. Refer to Eq. 2, where $\acute{S}$ is the retrieved secret data, Ex (.) is the extraction function, and $\bar{C}$ is the received Stego image at the receiver side, after being exposed to channel noise .[19]

$$S' = Ex(\quad \overline{C}), \tag{2}$$

### 2.3 Digital Image Steganography Properties, and Metrics.

Evaluation metrics are used to measure the imperceptibility, security, and capacity of the image steganographic methods. How well can a given method hide the secret message? The most commonly used metrics are described below.

#### 2.3.1 Imperceptibility

Since the primary method that exposes the existence of a secret message is the naked eye or the Human Visual System (HVS), imperceptibility is the main concern of stenographers. Digital image steganography imperceptibility, in short, means that the quality of the Cover image before and after embedding should be ideally identical such that any person would be incapable of distinguishing between them, whether visually or using statistical techniques. Therefore, when embedding a secret message in a Cover image, the sender should maintain the perception or the statistical mean of the Cover image unchanged .[19] There are two measuring tools that are widely used to evaluate the imperceptibility of steganographic systems, which are Peak Signal to Noise Ratio (PSNR) and Structural Index Similarity (SSIM).

PSNR of an image is calculated first by finding the Mean Square Error (MSE) between the Cover image and Stego image using Eq. 4(. Where $M$ and $N$ are the image height and width respectively, or image resolution, as it is also known as. $O$ is the number of the image channels. $I_(x, y, z_)$ is the pixel value of the Cover image at the $x$, $y$, and $z$ coordinates. And $I_(x, y, z'_)$ is the pixel value at $x$, $y$, and $z$ coordinates of the Stego image. Secondly, the logarithmic of the Mean Square Error is calculated to find PSNR. Refer to 3 where max is the maximum value of a pixel, and it can be calculated using $(2^n - 1)^2$, where n is the number of bits per pixel. It is worth noting that the higher the PSNR value, the higher the Stego image quality and lower the distortion as a result of the embedding function.

$$PSNR = 10log_{10}(\frac{max^2}{MSE}), \tag{3}$$

$$MSE = \frac{1}{MXNXO}\sum_{x=1}^{M}\sum_{y=1}^{N}\sum_{z=1}^{O}[(I_{(x,y,z)} - I'_{(x,y,z)})^2], \tag{4}$$

As previously mentioned, SSIM is another measuring tool for for evaluating the imperceptibility of Stego image. It has been published for the first time in the work of Zhou Wang et al. [39] as a new quality assessment method that is based on the degradation of structural information. The SSIM measuring mechanism relies on three main factors: luminance, contrast, and structure.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

$$SSIM = l(i, i`)C(i, i`)S(i, i`), \tag{5}$$

$$l(i, i`) = \frac{2u_i u_i` + C1}{u_i^2 + u_i'^2 + C1}, \tag{6}$$

$$C(i, i`) = \frac{2\Sigma_i \Sigma_i' + C2}{\Sigma_i^2 + \Sigma_i'^2 + C2}, \tag{7}$$

$$S(i, i`) = \frac{\Sigma_{ii}` + C3}{\Sigma_i \Sigma_i' + C3},, \tag{8}$$

The first factor, $l(i, i^{'})$ is a function that measures the difference between cover image $i$ and Stego image $i^{'}$ in terms of luminance. Where $u_i, u_i'$ are the luminance of the two images, and $C1$ is a constant used to prevent dividing by zero. $C1$ is recommended to have the following value.

$$u_i = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{O} i_{xyz}}{MNO}, \tag{9}$$

$$u_i' = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{O} i'_{xyz}}{MNO}, \tag{10}$$

$$C1 = (0.01x255)^2. \tag{11}$$

The second factor, $c(i, i^{'})$ measures the difference in contrast between the Cover image and the Stego image. Where $\Sigma_i, \Sigma_i'$ are the standard deviation of both the Cover image and the Stego image respectively. $C2$ is a constant that has the same purpose as $C1$, and is recommended to have the following value.

$$\Sigma_i^2 = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{O} (i_{xyz} - u_i)^2}{MNO}, \tag{12}$$

$$\Sigma_i'^2 = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{O} (i'_{xyz} - u_i')^2}{MNO}, \tag{13}$$

$$C2 = (0.03x255)^2. \tag{14}$$

Finally, the third factor $S(i, i^{'})$ compares the structure of the image before and after embedding. Where $\Sigma_i i^{'}$ is the covariance between Cover and Stego image, and $C3$ is a constant that serves the same purpose as $C1$ and $C2$. As before, $C3$ is recommended to have the following default value.

$$\Sigma_{ii}' = \frac{\sum_{x=1}^{M} \sum_{y=1}^{N} \sum_{z=1}^{O} (i_{xyz} - u_i)(i'_{xyz} - u_i')}{MNO}, \tag{15}$$

$$C3 = \frac{C2}{2} \tag{16}$$

The fore mentioned three factors, using the SSIM tool, are highly sensitive to any distortion caused by embedding, as Setiadi et al. stated .[32] If all of these three factors are at their maximum, which is 1, then SSIM is maximized, meaning that the two images being compared are effectively identical. Ideally, having SSIM result as 1 is the target of any steganographic system. However, such a case is infeasible in any real life scenario. Therefore, the goal for highly imperceptible steganography is to come as close to 1 as possible.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

### 2.3.2 Security

Security is a major concern for any steganographic method. According to the definition provided by ,[19] security basically means the change caused by the embedding function is undetectable (i.e., unnoticeable by any steganalytics). The security of steganographic schemes is typically evaluated by testing their resistance to well-known steganography tools. The result of this kind of investigation is usually reported in terms of accuracy and detection, or classification errors made by the Steganalyzer.[19,35]

A so-called Steganalyzer is a classification tool or a model that takes an input image and outputs a prediction that defines whether this image is predicted as the Cover versus the Stego. This Steganalyzer could be feature-based, such as SRM_EC ,[3] and ATS, or CNN-based, such as Xu_Net [40] and GBRAS_Net .[28]

Accuracy is defined as the total number of correct predictions made by the Steganalyzer across all classifications. It is computed using four terms used to calculate accuracy: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True positive and true negative are the model's correct predictions, whereas false positive and false negative are its errors. A lower level of accuracy indicates a higher level of security. Refer to the table 1 .[19,35]

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{17}$$

Detection error rate is another security metric, defined as the total number of incorrect predictions made by the Steganalyzer across all classifications. A higher level of detection error indicates higher level of security.

$$Detection_Error = \frac{FP + FN}{TP + FP + FN + TN}, \tag{18}$$

### 2.3.3 Capacity

One of the major challenges in steganography is having a balance between the capacity of image steganography and imperceptibility (and security as mentioned in section 2.3.2). The goal, embed more data inside the image(s) without sacrificing the imperceptibility or security of the secret message(s) and any concomitant risks. The two main metrics that are used to calculate image steganography capacity are *relative capacity* and *payload capacity*. The relative capacity basically translates to the amount of hidden capacity with respect to image size and is defined as the ratio between the absolute capacity (i.e., the hiding capacity per image) and the size of the Cover image. Payload capacity, on the other hand, is defined as the number of bits embedded in each pixel.[19,35]

$$Relative_Capacity = \frac{AbsoluteCapacity}{imagesize}, \tag{19}$$

$$Payload_Capacity = rx8x3(bpp), \tag{20}$$

$$r = 1 - \frac{1}{WxH} \sum_{x=1}^{W} \sum_{y=1}^{H} [i_{x,y} - i'_{x,y}], \tag{21}$$

## 3 Content-based Method Summary

As previously stated, *content adaptive steganography* attempts to embed secret messages into a lessor detectable area of the Cover image. The locality is determined by adjusting the embedding location by searching for the best (most imperceptible) locality within the Cover image content. This goal is achieved in the spatial domain by following different approaches, which have been grouped into three categories namely, CNN-based, GAN-based, and minimizing additive distortion function-based.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

### 3.1 CNN-based

One of the approaches to producing content-adaptive image steganography has the advantage of deep learning architecture. The advantage utilizes the superior capability of Neural Networks (NN) to generate their own model that demonstrates the relationship between the Cover image and Stego image rather than heuristically designing this relation. This approach utilizes an encoder-decoder architecture. The encoder works as the embedding algorithm. While the decoder works as the extraction algorithm. Refer to Figure 1. Several works have been published based on this principle. They only differ on their architecture pertaining to the number of filters used, strides, filter size, activation function, and loss function used. Furthermore, the Cover image and the secret message concatenation process are another obvious difference, in addition to their sizes and types.

At the beginning, researchers utilized existing object detection architectures such as SSD ,[23] R-FCNN ,[2] and Faster R_CNN [29] to obtain the most complex texture region in the cover image .[15] Subsequently, steganographic algorithms are used, including WOW '[13] , HUGO ,[25] and S-Uniward [14] to embed secret information in the selected area(s) of the image .[24] In 2019, Duan et al [4] proposed a reversible encoder decoder architecture. Their encoder was U_Net, and their decoder was a combination of six convolution layers with a filter size of 3 x 3. As previously stated, what distinguishes their work is its reversible nature. Which means, on the receiver side, the primary concern is not only the retrieval of the secret message without errors but also the Cover image. Duan et. al., in their proposed architecture was able to accomplish a relative capacity of 1. With the help of the FC_DenseNet56 [17] encoder, he was able to maintain the same relative capacity simultaneously with higher imperceptibility .[6]

In 2020, Duan et. al., [5] their experiments showed their methods doubled relative capacity. They compared the performance of three encoder architectures: conventional CNN, U_Net, and FC_DenseNet encoders to hide two RGB images inside one cover image. The comparison held in terms of model size, convergence speed, and average PSNR. They conclude that the FC_DenseNet encoder was superior. The decoder network used by [4,6] was the same as the one used by .[5] Instead of concatenating the cover and secret images, similar to,[4–6] before input to the encoder. Rahim et al [27] tried something different. They input each image, cover and secret, to a separate, but parallel part of the encoder and concatenating the output of each parallel layer along the way. In consequence, Rahim et al.'s encoder was distinguished by these two parallel architectures. The concatenated feature maps are used to create the Stego image. Rahim et al.'s contribution was not limited to this unique architecture. They also introduced a new loss function that ensures joint end-to-end training of encoder-decoder networks.

All previously summarized architectures have their Cover image in RGB format. Zhang et al.'s '[43] basic model, which follows the encoder-decoder architecture, uses the YCrCb cover image format. They converted the RGB cover image into YCrCb format. Since the semantic and colour information are present in the Cr and Cb channels, they decided to embed only in the Y channel, by input to (only) their encoder architecture. Secret images are also converted from RGB format to gray scale format as a preprocessing step. The encoder architecture developed by Zhang et. al., benefited from the Inception block (a building block for convolutional neural networks that efficiently extracts features at multiple scales using parallel convolutional layers of different sizes). Moreover, they constructed a mixed loss function that takes the advantages of weighted SSIM, MSE, and MS-SSIM metrics to properly fit the human visual system (HVS). Table 2 summarizes the review on the CNN-based adaptive image steganography approach.

### 3.2 GAN-based

The Generative Adversarial Network (GAN) is a class of unsupervised machine frameworks designed by Lan Goodfellow and his colleagues in 2014 .[11] Generally, GANs consists of two neural network models competing against one another in a min-max game. One model is known as a generator, and the other is known as a discriminator. The generator plays the role of a forger who aims to generate data similar to the original (real data) from noise, so that the investigator or the discriminator does not notice its artificiality (i.e., falsity). In short, the job of the generator is to generate fake data that is similar to the real data, while the job of the discriminator is to distinguish between the fake data and the real data. The generator tries to learn the distribution of the real data, and the discriminator tries to learn the boundaries between the real and fake data, or classes. In 2015, Radfoed et. al., [26] extended the GAN approach to deep convolutional networks (DCGAN), where both the Generator (G) and Discriminator (D) adopted the CNN architecture.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

Table 1: Confusion Matric for calculating accuracy, and classification error.

| **Actual** | **Predictions** | |
|---|---|---|
| | **True** | **False** |
| **True** | True Positive | False Negative |
| **False** | False Positive | True Negative |

Table 2: Summary of CNN-based image steganography evaluation metrics.

| **R** | **Image** | **SSIM** | **PSNR** | **P.C** | **R.C** | **Security** | **Dataset** | **Architecture characteristics** |
|---|---|---|---|---|---|---|---|---|
| [5] | RGB | 0.981 | 35.852 | 47.6 bpp | 2 | ROC graph: Indicate certain degree of anti-steganalysis ability. | ImageNet, Remote sensing, and aerial images. | FC_DenseNet |
| [27] | RGB | 0.96 | 33.7 | 8 bpp | 1 | - | LFW, ImageNet, PASCAL_VOC12. | - |
| [4] | RGB | 0.9728 | 39.9837 | 23.96 bpp | 1 | - | ImageNet | U_Net |
| [6] | RGB | 0.985 | 39.556 | 23.96 bpp | 2 | ROC graph: Indicate certain degree of anti-steganalysis ability. | ImageNet. | FC_DenseNet56 |
| [1] | Grayscale image hidden in B channel of RGB image | - | 36.97 | 8 bpp | 1/3 | Xu_Net Steganalyzer Accuracy: 0.7682. | ImageNet. | - |
| [43] | Grayscale image hidden in RGB image | 0.9534 | 34.57 | 8 bpp | 1/3 | Accuracy of CNN-based steganalysis model on tiny dataset was 0.7814. | LFW, ImageNet, PASCAL_VOC12. | Inception block[1] |

*DOI: https://doi.org/10.54216/FPA.180216*
*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

Researchers noticed that the GAN architecture simulates steganography and steganalysis competitions. Accordingly, they adopted the GAN paradigm as a content adaptive image steganography second approach, using a variety of different architectures. Moreover, Volkhonskiy et. al., ,[38] generated their own steganography GAN block diagram known as the steganographic generative adversarial network (SGAN). The paradigm of the GAN-based framework consists of four components: Generator G, Discriminator D, Steganalysis S, and Information Embedding. Generator G takes noise as input to produce a Fake image that looks real to the discriminator D. The information embedding block embeds the secret message, usually text, in the generated fake image to produce Stego images. Both the Fake and Stego are input to the Steganalyzer S. SGAN separates image generation and secret message embedding into two stages. Additionally, the receiver task is not simulated clearly, which were the two reasons behind the motivation for proposing Steganographic Encryption Generative Adversarial Network (SEGAN) .[16]

SEGAN contains only three components: Generator G, Encryptor (i.e., Alice), taking input noise, secret message, and a secret key to produce the Stego (i.e., Fake) image. Discriminator D (i.e., Eve) works trying to classify images as real or fake. Decrypt (i.e., Bob) tries to gain the secret message back from the generated Stego image using the secret key. Therefore, SEGAN better simulates Alice, Bob, and Eve prisoner's problem defining steganography, as compared to SGAN. However, similar to SGAN, SEGAN also generates fake, or Stego images from noise input, which in turn makes the discriminator and Steganalyzer performance rely on the secret message embedding location and the generator's ability to learn the real image distribution and then successfully converting the noise to fake images.

Hayes et al.[12] and Shi et. al., [33] made the generator's job easier by sending the real Cover image as input to the generator directly, in addition to the secret message. This modification makes possible that the generator can focus on the steganographic main task while embedding in a less detectable areas simultaneously. Thus, the HayersGAN [12] architecture has been adopted by various researchers with small scale modifications. Before proceeding through those modifications, what follows are some naming conventions.

- Generator G, Encoder, Alice: the sender who is trying to generate a Stego image from a Cover image and secret message.;
- Decoder, Bob: the receiver who is trying to get the secret message back.;
- Steganalyzer, Discriminator, Critic, Eve: the third part who is trying to figure out the existence of a secret message in the cover image.

Zhu et al.[45] proposed HIDDeN, a GAN-based method with four main components: encoder, decoder, discriminator, and noise layer. The encoder receives the cover image and the secret message as inputs and generates an encoded image, which then is passed to the noise layer to generate the noised image. The decoder decodes the secret message from the image with noise, while the discriminator determines the likelihood that a given image is encoded. The noise layer is added to enhance the encoded image robustness.

Zhang et al. [42] proposed SteganoGAN, a GAN model with an encoding network for creating Stego images and a decoding network for extracting the secret message from the Stego image. The encoder embeds the secret message within the cover image, while the decoder recovers the secret message, and a critic, or Steganalyzer, evaluates the quality of the images generated by generating a score. For different payload capacities, the basic, residual, and dense versions of the encoder are also talked about.

HayersGAN ,[12] HiddenGAN ,[45] and SteganoGAN [42] all use images to hide secret messages, not secret images. ISGAN [43] and ChenGAN [1] do indeed conceal secret images. Zhang et al. were able to embed a gray scale image within the Y channel of a YCrCb format cover image by incorporating a Steganalyzer network into their encoder decoder architecture or basic model .[43] Similarly, ChenGAN is an extension of a basic model that embeds a gray scale secret image in the B channel of an RGB cover image. According to the fact that the human visual system (HVS) is less sensitive to colour variation in the B channel than the R and G channels .[1] Table 3 summarizes the review on GAN-based adaptive image steganography approach.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

Table 3: Summary of GAN-based image steganography evaluation metrics.

| **R** | **Image** | **SSIM** | **PSNR** | **P.C** | **Security** | **Dataset** | **Architecture characteristics** |
|---|---|---|---|---|---|---|---|
| [1] | Gray scale image hidden in B channel of RGB image | - | 37.35 | 8 bpp | Xu_Net Steganalyzer Accuracy: 0.7248 Detection error: 0.2752 | ImageNet | - |
| [43] | Gray scale image hidden in RGB image | 0.968 | 34.63 | 8 bpp | Accuracy of CNN-based steganalysis model on tiny dataset: 0.7360 | LFW, ImageNet, PASCAL_VOC12 | Inception block |
| [45] | Hiding message into image | - | - | 0.203 bpp | ATS Steganalyzer Accuracy: 0.02 Detection error: 0.98 | COCO, BOSS | - |
| [12] | Hiding message into image | - | - | 0.4 bpp | ATS Steganalyzer Accuracy: 0.95 Detection error: 0.05 | celebA, BOSS | - |
| [42] | Binary message into RGB image | 0.90 | - | 4.4 bpp | auROC=0.59 FP= Stego as Cover TP= Stego as Stego | COCO, Div2k | DenseNet |

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

### 3.3 Minimizing Additive Distortion Function-based

The third approach utilizes the notion of minimizing the additive embedding distortion and cost assignment function to derive a new content-adaptive image steganography method. Instead of training one model, generator, or encoder to adapt the embedding location to cover image content, this approach divides the task into two subtasks. The first task utilizes a well-designed embedding distortion, or cost assignment, function to generate a cost matrix for each cover image. The second task makes use of a coding schema such as Syndrome Trellis Codes (STC), which takes as input, a cover image and its corresponding cost matrix, as well as a secret message, and produces an output Stego image.[8] This is combined with minimal cost or minimal embedding distortion. It is worth noting that the first task is what controls how the embedding content is adaptive.

Minimizing embedding distortion simply means minimizing a well-designed additive distortion function, which is defined in Eq.22.

$$D(X,Y) = \sum_{i=1}^{W} \sum_{j=1}^{H} p_{i,j}[x_{i,j} - y_{i,j}], \tag{22}$$

Where $D(X,Y)$ is the measure of the additive distortion caused by changing Cover image $X$ to Stego image $Y$. $H$ and $W$ are the height and width of the Stego and Cover image, respectively. $p_i, j$ is the cost, or probability of changing pixel $x_i, j$ in Cover image to $y_i, j$. P is a matrix representing the cost of changing or probability of changing pixel $x_i, j$ to $y_i, j$. The cost and the probability of change are inversely related.

Researchers noticed that an accurate assignment of the cost of embedding has a very significant influence on the performance of embedding. For example, giving texture regions low embedding costs and smooth regions high embedding costs in the right way improves the whole embedding process and reduces the distortion cost. Therefore, they have developed a cost function, (i.e., distortion function) whose main task is to quantify the effect of making changes to a pixel.

Initially, the cost function was heuristically designed using hand-crafted methods such as highly undetectable Stego HUGO ,[25] wavelet obtained weights WOW ,[13] high pass, low pass, and low pass HILL ,[21] spatial universal wavelet retrieval distortion S_UNIWARD ,[14] and minimizing the power of optimal detector MiPOD .[31] The previously handcrafted distortion function achieved an acceptable level of security. However, a main drawback is that the detectability factor was not taken into consideration when designing the cost function. The cost of embedding is related to detectability, according to Pevny T .[25] However, it had not been taken due to practical difficulties.

With the discovery of GAN, simulating the distortion and detectability relationship, which was impossible before, became possible. Tang et al. [37] were the first to utilize GAN to design a distortion function automatically. Tang et al [37] proposed in 2017, Automatic Steganographic Distortion Learning using a Generative Adversarial Network (ASDL_GAN). It consists of three components: Generator G, Ternary Embedding Simulator (TES), and Discriminator D. The generator takes as input, a Cover image, and target capacity for which change probability is to be produced. Recall that the change probability matrix is the inverse of the change cost matrix. Their proposed generator consists of 25 groups. The initial 24 groups consist of a 7x7 convolution kernel, a Batch Normalization (BN), and a rectified linear unit (ReLu) activation layer. A short-cut connection is used after every two-group. The same operations are carried out in the 25th group, except that the Sigmoid is used as the activation function on top of BN. After the Cover image passes the 25th group, multiplication by 0.5 and ReLu are performed to ensure that the probability matrix values, which have the same size as the Cover image, lie within (0,0.5) and leave no potential security holes. The TES takes as input the probability map matrix output by the generator and a matrix of floating-point numbers that simulates the secret message and outputs a modification map, which has the same size as the input image. Refer to Eq.23. where $n_i, j$ is the floating-point value, $p_i, j$ is change probability value, $m_{i,j}$ is modification value at a certain pixel. The modification map is added to the cover image to output a Stego image. Therefore, TES's main task is to simulate the stair case function for the ternary embedding operation, which modifies any pixel $x_i, j$ to $y_{i,j}$ either by $x_{i,j} + 1$, $x_{i,j} - 1$, or $x_{i,j}$.

$$m_{i,j} = \begin{cases} -1, \text{ if } n_{i,j} < \frac{p_{i,j}}{2} \\ 1, \text{ if } n_{i,j} < 1 - \frac{p_{i,j}}{2} \\ 0, \text{otherwise} \end{cases}, \tag{23}$$

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

The fact that the stair case function defined at Eq. 23 or the ternary embedding operation is not differentiable and does not preserve the gradient loss during back propagation was the motivation behind proposing TES by Tang et al., which was itself a mini-neural network. The discriminator takes an image as input, which could be a Cover or Stego image, and outputs a prediction probability. The Discriminator is based on the Xu_Net architecture.

Even though Tang et al. were able to design a distortion function automatically for the first time, ASDL_GAN security performance was inferior to hand-crafted distortion functions. Therefore, Yang et al. [41] decided to make several improvements over the work of Tang et al. Firstly, changing the TES from a mini-neural network to a Tanh_simulator as an attempt to avoid the long pre-training time required by the mini-neural network Secondly, utilizing the pixel-wise segmentation capability of the U-NET, by proposing a U_Net based generator, which contributes to increasing security performance and decreasing training time. Thirdly, they enhances the discriminator design by incorporating the generator's output probability map to boost resistive performance against selection-channel awareness (SCA) based steganalysis methods. UT-SCA-GAN is superior to ASDL_GAN in terms of security and training speed.

Leveraging the benefit of GAN to design the distortion function automatically requires utilization of two loss functions, namely the discriminator loss and generator loss. The Discriminator loss is used to ensure that the Discriminator is able to properly distinguish between the Cover and Stego image. The Generator loss was made up of the entropy loss and the adversarial loss. The entropy loss ensures that the target embedding capacity was met, and the Adversarial loss makes it harder to find the code.

Minimizing additive distortion function-based image steganography performance is usually computed in terms of security with respect to payload capacity. In this case, the performance does not consider imperceptibility because it's guaranteed by default. Security is usually computed by detection or classification errors made by the Steganalyzer. SRM_EC, max-SRM_EC, and Xu_Net are the most commonly used Steganalyzers. Table 4 summarizes the reviewed distortion function security performance using SRM_EC.

## 4 Analysis and Discussion

An analysis of the three different approaches to content adaptive image steganography in terms of advantages, disadvantages, and the gap left for improvement is summarized in table 5.

The main advantage of the CNN-based approach over other reviewed approaches is the high capacity. Researchers were able to embed a maximum relative capacity of 2. Moreover, this CNN-based approach has a moderate architecture if compared to other approaches in terms of complexity. Only the encoder and the decoder networks are needed.

However, this moderately complex architecture makes it impossible to incorporate security factors or a Steganalyzer. When training the encoder, encoder weights are usually updated based on imperceptibility loss. Even though the model is trained using the imperceptibility loss and its correlates to security, as shown in table 1, this approach suffers from low imperceptibility and security. This low imperceptibility and security are due to two reasons, i) high embedding capacity and ii) the architecture itself.

Recall that in a CNN-based architecture, the encoder consists of an encoder, which performs embedding, and a decoder, which performs extraction. The training for those two models is separate. However, according to Subramanian et al., perfect image steganography requires two models to be trained end-to-end under the same training circumstances, meaning that the sender and receiver models are interconnected, and one model's loss may affect the other model.

Therefore, the GAN-based approach outperforms the CNN-based approach by having end-to-end training and incorporating a detectability factor when training, which in turn makes high security one of its main strengths. Refer to table (2). The decrease in accuracy value achieved when incorporating Steganalyzer models into Zhang et al. and Chen et al. basic models clearly shows this [12,13]. However, a problem that resulted is the effect of increasing complexity, which in turn makes the training and GAN convergence failure one of the main challenges [5].

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

In general, Comparing the security of GAN-based approach to minimizing additive distortion function-based approach is not fair. The reason for that is the differentiation of the evaluation metrics, in addition to the variation in capacity. This is considered the second challenge faced by researchers when proposing steganographic methods.

However, Hayes et. al., attempted to compare their security with distortion functions-based security using the state-of-the-art Steganalyzer banded on Artificial Training Sets (ATS). HayersGAN achieved an accuracy of 0.83 using the BOSSbase dataset. This accuracy is considered worse than the HUGO, WOW, and S-UNIWARD distortion functions for the same capacity.

It is worth observing that even though the GAN-based approach embeds lower relative capacity than the CNN-based approach, the SSIM and PSNR values of GAN-based approach are less than or equal to the CNN-based proposed work. Refer to table 2, and 3. Minimizing additive distortion embedding based on a well designed distortion function makes high imperceptibility a guarantee. Moreover, the automatically designed distortion function encounters a detectability factor. Therefore, the main drawback of GAN-based approach is the low capacity.

Briefly, each one of the reviewed content adaptive image steganography approaches have their pros and cons. Moreover, the choice of which one to utilize is based on which factor is to be prioritized given a certain application.

## 5 Conclusion

Image steganography is the method of transmitting confidential information by concealing it within a Cover image. Content-adaptive image steganography is a promising new trend in the steganographic field which adapts the embedding location based on the cover image features. The review of all relevant works in content-adaptive image steganography led to their classification into three categories: CNN-based, GAN-based, and minimizing additive distortion function-based.

Research in steganography has utilized methods of deep learning to develop innovative yet complex encoder-decoder architectures. The encoder is responsible for embedding at the sender side, and the decoder is responsible for extraction at the receiver side. The CNN network is used and trained on each side. The development of GAN has provided the possibility of incorporating the Steganalyzer into the encoder-decoder network and making training interconnected.

Minimizing the additive distortion function-based simplified GAN-based architecture by splitting the embedding task into two subtasks and has further lead to an innovative design for the distortion function of either automatically or heuristically.

Each one of the reviewed approaches provides insight into their own advantages and disadvantages. The choice of which one to use depends mostly on the property which is of most importance given the application priorities. As was carefully discussed and critiqued some of the decision metrics include security, imperceptibility, and/or capacity.

We believe that the future is trending toward a focus on utilizing GAN architectures because they can *perfectly* simulate the embedding process. As well, the pixel-wise segmentation networks such as FC_DenseNet and U_Net are sufficient as encoders, senders, or generator networks. These segmentation networks can be easily Incorporated into GAN architectures.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

Table 4: Summary of minimizing additive distortion function-based image steganography evaluation metrics. Security evaluation based on detection error made by steganalysis (SRM_EC), BOSSBase dataset, 5,000 SRM_EC training images, and 5,000 SRM_EC testing images.

| R | 0.1 bpp | 0.2 bpp | 0.3 bpp | 0.4 bpp | Training Dataset |
|---|---|---|---|---|---|
| HILL[21] | 0.4519 | 0.3840 | 0.3248 | 0.2782 | - |
| S_UNIWARD[13] | 0.4229 | 0.3389 | 0.2741 | 0.2197 | - |
| MiPOD[31] | 0.4229 | 0.3389 | 0.2741 | 0.2197 | - |
| ASDL_GAN[41] | 0.3702 | 0.3207 | 0.2704 | 0.17 | SZUBase |
| UT_Net_GAN[8] | 0.4414 | 0.3852 | 0.3363 | 0.2911 | SZUBase |

Table 5: Summary of content adaptive image steganography approaches' advantages, disadvantages, and gaps for improvement.

| Approach | Advantage | Disadvantage | Gaps Left for Improvement |
|---|---|---|---|
| **CNN-based** | • Moderate complexity.<br>• High capacity. | • Low security.<br>• Low imperceptibility. | • Increase imperceptibility or security with respect to capacity.<br>• Increase capacity. |
| **GAN-based** | • High security. | • High complexity.<br>• Low imperceptibility. | • Increase imperceptibility or security with respect to capacity.<br>• Increase capacity. |
| **Minimizing Additive Distortion Function - Heuristically Designed** | • Low complexity.<br>• High imperceptibility. | • Low capacity.<br>• Does not simulate correlation between security and image distortion. | • Increase security with respect to capacity. |
| **Minimizing Additive Distortion Function - Automatically Designed** | • Moderate complexity.<br>• High imperceptibility.<br>• Simulates correlation between security and image distortion. | • Low capacity. | • Increase security with respect to capacity. |

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

**References**


[1] Beijing Chen, Jiaxin Wang, Yingyue Chen, Zilong Jin, Hiuk Jae Shim, and Yun-Qing Shi. High-capacity robust image steganography via adversarial network. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(1):366–381, 2020.

[2] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[3] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 48–53. IEEE, 2014.

[4] Xintao Duan, Kai Jia, Baoxia Li, Daidou Guo, En Zhang, and Chuan Qin. Reversible image steganography scheme based on a u-net structure. *IEEE Access*, 7:9314–9323, 2019.

[5] Xintao Duan, Nao Liu, Mengxiao Gou, Wenxin Wang, and Chuan Qin. Steganocnn: image steganography with generalization ability based on convolutional neural network. *Entropy*, 22(10):1140, 2020.

[6] Xintao Duan, Liu Nao, Gou Mengxiao, Dongli Yue, Zimei Xie, Yuanyuan Ma, and Chuan Qin. High-capacity image steganography based on improved fc-densenet. *IEEE Access*, 8:170174–170182, 2020.

[7] Xintao Duan, Haoxian Song, Chuan Qin, and Muhammad Khurram Khan. Coverless steganography for digital images based on a generative model. *Computers, Materials & Continua*, 55(3):483–493, 2018.

[8] Tomáš Filler, Jan Judas, and Jessica Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, 2011.

[9] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on information Forensics and Security*, 7(3):868–882, 2012.

[10] Despoina Giarimpampa. Blind image steganalytic optimization by using machine learning, 2018.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. *Advances in neural information processing systems*, 30, 2017.

[13] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, pages 234–239. IEEE, 2012.

[14] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1–13, 2014.

[15] Israr Hussain, Jishen Zeng, Xinhong Qin, and Shunquan Tan. A survey on deep convolutional neural networks for image steganography and steganalysis. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(3):1228–1248, 2020.

[16] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

[17] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

[18] Neil F Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998.

[19] Inas Jawad Kadhim, Prashan Premaratne, Peter James Vial, and Brendan Halloran. Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research. *Neurocomputing*, 335:299–326, 2019.


*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

[20] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[21] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4206–4210. IEEE, 2014.

[22] Wen-Nung Lie, TC-I Lin, and Chia-Wen Lin. Enhancing video error resilience by using data-embedding techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(2):300–308, 2006.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[24] Ruohan Meng, Steven G Rice, Jin Wang, and Xingming Sun. A fusion steganographic algorithm based on faster r-cnn. *Computers, Materials & Continua*, 55(1):1–16, 2018.

[25] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *International workshop on information hiding*, pages 161–177. Springer, 2010.

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[27] Rafia Rahim, Shahroz Nadeem, et al. End-to-end trained cnn encoder-decoder networks for image steganography. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[28] Tabares-Soto Reinel, Arteaga-Arteaga Harold Brayan, Bravo-Ortiz Mario Alejandro, Mora-Rubio Alejandro, Arias-Garzon Daniel, Alzate-Grisales Jesús Alejandro, Burbano-Jacome Alejandro Buenaventura, Orozco-Arias Simon, Isaza Gustavo, and Ramos-Pollan Raul. Gbras-net: a convolutional neural network architecture for spatial image steganalysis. *IEEE Access*, 9:14340–14350, 2021.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[30] David L Robie and Russell M Mersereau. Video error correction using steganography. *EURASIP Journal on Advances in Signal Processing*, 2002(2):1–10, 2002.

[31] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2015.

[32] De Rosal Igantius Moses Setiadi. Psnr vs ssim: imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, 80(6):8423–8444, 2021.

[33] Haichao Shi, Xiao-Yu Zhang, Shupeng Wang, Ge Fu, and Jianqi Tang. Synchronized detection and recovery of steganographic messages with adversarial learning. In *International Conference on Computational Science*, pages 31–43. Springer, 2019.

[34] Alan Siper, Roger Farley, and Craig Lombardo. The rise of steganography. *Proceedings of student/faculty research day, CSIS, Pace University*, 2005.

[35] Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. Image steganography: A review of the recent advances. *IEEE access*, 9:23409–23423, 2021.

[36] Shunquan Tan and Bin Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Signal and information processing association annual summit and conference (APSIPA), 2014 Asia-Pacific*, pages 1–4. IEEE, 2014.

[37] Weixuan Tang, Shunquan Tan, Bin Li, and Jiwu Huang. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, 24(10):1547–1551, 2017.

[38] Denis Volkhonskiy, Boris Borisenko, and Evgeny Burnaev. Generative adversarial networks for image steganography. 2016.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*

[39] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[40] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016.

[41] Jianhua Yang, Kai Liu, Xiangui Kang, Edward K Wong, and Yun-Qing Shi. Spatial image steganography based on generative adversarial network. *arXiv preprint arXiv:1804.07939*, 2018.

[42] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019.

[43] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. *Multimedia tools and applications*, 78(7):8559–8575, 2019.

[44] Wei Zhang, Sen-Ching S Cheung, and Minghua Chen. Hiding privacy information in video surveillance system. In *IEEE International Conference on Image Processing 2005*, volume 3, pages II–868. IEEE, 2005.

[45] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018.

*DOI: https://doi.org/10.54216/FPA.180216*

*Received: August 14, 2024 Revised: November 24, 2024 Accepted: January 30, 2025*