Journal of Theoretical and Applied Information Technology
30th April 2025. Vol.103. No.8
© Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL METHOD FOR FLEXIBLE CAPACITY MULTI-IMAGE HIDING

HESHAM F.ABDELRAZIK[1], AHMED S.ELSAYED[2], SALLY S.ISMAIL[3], ABEER M.MAHMOUD[4]

[1]Demonstrator, Computer Science Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt
[2,3]Lecturer, Computer Science Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt
[4]Professor, Computer Science Department, Faculty of Computer and Information Science, Ain Shams University, Cairo, Egypt

E-mail: [1]hesham.fathy@cis.asu.edu.eg, [2]ahmed_salah@cis.asu.edu.eg, [3]Sallysaad@cis.asu.edu.eg, [4]abeer.mahmoud@cis.asu.edu.eg

ABSTRACT

The noteworthiness of multiple image-hiding has increased in the current digital era. This owes to the growing demand for robust data protection and secure communication. The success of the multiple image hiding algorithm depends on concealing multiple secret images in one cover image and perfectly recovering all secret images at the receiving end. Unfortunately, the two primary challenges with current multi-image hiding methods are preserving high capacity while reducing visual distortions and guaranteeing precise recovery of every hidden image. The main challenge is to keep the hidden image visually obscured while concealing as much information as possible from a single cover image. In this paper, a novel framework for concealing multiple images using an invertible hiding neural network and image super-resolution is proposed. Extensive experiments on DIV2K, ImageNet, and COCO demonstrated that the proposed framework obtained significant results in terms of both imperceptibility and recovery accuracy of the secret image. Achieving an average of 27.16 dB, 26.65 dB, and 32.13 dB of PSNR and 0.810, 0.783, and 0.885 of SSIM for hiding four secret images with upscaling factor 2, respectively. The proposed framework successfully concealed and revealed up to 64 secret images. These results confirm the effectiveness of the proposed method in achieving both high invisibility and enabling multiple image concealment with a large capacity.

Keywords: *Image hiding, Image rescaling, Invertible hidden neural network, Image super-resolution, Deep learning.*

1. INTRODUCTION

With easy access to the Internet and the increasing popularity of social media platforms, this has resulted in widespread sharing of information online. Multimedia digital communication (text, audio, images, and videos) has been widely used [1,2]. Variant risks exist during delivering sensitive messages despite recent advances in this domain. For example, when transferred across an unsafe network channel, they are vulnerable to theft and ease of access. Therefore, digital communication is beneficial in several sectors. As all data are exchanged over the Internet, transmitting them from the point of origin to the destination calls for exceptional security [3,4].

This presents a significant problem for researchers focusing on protecting and preserving data privacy. There are numerous strategies to protect against data theft and hacking. Figure 1. Shows overview of data security techniques (Digital Watermarking, Steganography and Cryptography). Steganography is classified based on multimedia (text, audio, images, or videos). Image steganography is categorized into traditional and deep-learning techniques.

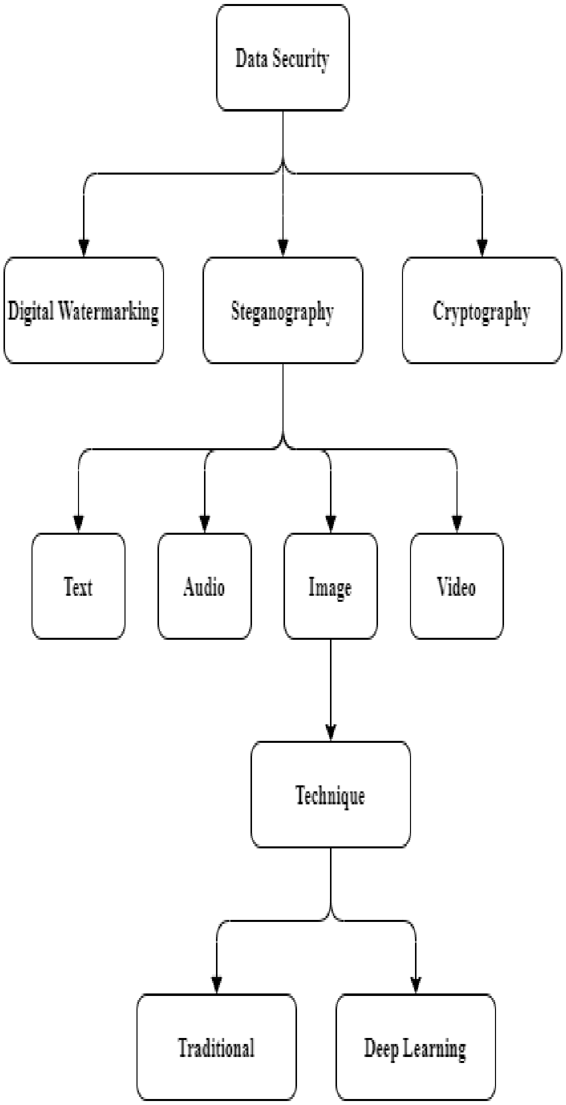


Figure 1: Classification Of Data Security Techniques For Information Hiding.

As shown in Figure. 1, these methods frequently use cryptography [5], steganography [6], and digital watermarking [7,8] technologies. First, digital watermarking inserts code or marks into digital content to assist in securing copyright. Cryptography mixes data, which makes it more difficult for outsiders to decipher. Therefore, data are rendered unbreakable and incomprehensible while remaining visible to humans, making it more difficult for hackers to obtain sensitive data [9,10]. Unlike the other two methods, steganography allows for the concealment of a secret message from other types of data, such as texts, images, or videos. Using image steganography, one can create a stego image—a hidden image that is only visible to those who are aware of its existence—by hiding a secret image within a cover image. The stego image must often be identical to the cover image, owing to security considerations.

Existing multi-image hiding techniques face challenges such as limited capacity, visual distortions, and poor recovery quality, making them unsuitable for high-security applications requiring robust data protection. Its applications encompass vital areas such as secret communication and privacy protection. Using multiple image-hiding techniques not only increases the capacity for safe data transfer but also provides a workable solution for situations in which secure data dissemination and covert data storage are required. However, concealing several images is challenging. Due to a decrease in resolution and possible loss of important details. Therefore, the essential priority is to preserve the visual quality of hidden images. To overcome these limitations, this paper proposes a novel framework that integrates an invertible hidden neural network (IHNN) and image super-resolution (SR) to enhance hiding capacity while ensuring high invisibility and accurate image retrieval. To the best of our knowledge, no study has explored a combination of invertible hidden neural networks and image super-resolution in image-hiding tasks.

The main contributions of this study are summarized as follows:

- To conceal several hidden images into a single cover image, a novel framework based on invertible hidden neural networks and super-resolution is proposed.
- A comprehensive summary of various multiple image hiding techniques is presented.
- Extensive experiments are conducted to prove the effectiveness of the proposed approach.

The remainder of this paper is organized as follows. Related works are discussed in Section 2. The overall approach and the proposed method are discussed in detail in Section 3. The experimental results and an extensive analysis of the overall approach are presented in Section 4. Finally, Section 5 provides a conclusion and future work.

2. RELATED WORK

Although image steganography techniques have existed for a while, the rise in data traffic over the Internet has made them increasingly important [1,6]. Deep learning is now widely employed in image steganography because of the abundance of available data [3]. Despite being a recent area of research, image steganography has benefited from deep learning techniques such as the use of convolutional neural networks (CNNs) [11,12] and generative adversarial networks (GANs) [13,14]. This section's primary objective is to present recent state-of-the-art studies relevant to our proposed work. Table 1 shows the comprehensive summary of various multiple image-hiding techniques. Most state-of-the-art studies used the common datasets DIV2K [33], COCO [31], and ImageNet [28]. Achieved reasonable results with an average peak signal-to-noise ratio (PSNR) greater than 30 dB and a Structural Similarity Index (SSIM) greater than 0.870. However, up to four images were hidden.

Ashwaq Alabaichi et al. proposed an approach for spatial-domain image steganography employing a secret map and LSB techniques. The main source of inspiration for the secret maps was 3D chaotic maps, including 3D Chebyshev and 3D logistic maps. The foundation of this technique is the idea of selecting a pixel from a host image and performing random insertion. Secret keys are used to safeguard hidden information on a secret map. An embedding capacity of 8 bits per pixel (bpp), average PSNR of 39.14 decibels (dB), and average SSIM of 0.996 were the results of testing over 30 color images taken from well-known standard images[15].

Biswas R and Bandyapadhay SK proposed an approach for frequency-domain image steganography that has been utilized with a Genetic Algorithm (GA) using multibit LSB steganography. The multi-bit LSB approach was utilized to augment the payload with a genetic algorithm (GA) employed for optimization. Additionally, a pseudo-random function was utilized to sample pixels through a discrete cosine transform (DCT) to enhance robustness. They used at least 2000 images from various standard image sources for evaluation and achieved an average embedding capacity of 2.0 bpp, and average PSNR of 38.27 dB [16].

Baluja [17] proposed an autoencoder architecture-based deep neural network as a single model aimed at embedding a full-size color image within another identically sized image. Their model consisted of three layers: the prep network, hidden network, and reveal network. Before being introduced as input to the hiding network, which utilizes a cover image and the output of the preparation network to generate the container image, the secret image is prepared via the prep network. Uncovering the cover image allows the revealing network to decode the secret image from the container image. The system achieved an average PSNR of 34.13 dB and average SSIM of 0.925 on natural images randomly selected from the ImageNet database [28].

Ray et al. proposed a novel edge detection-based steganography technique that maximizes payload while minimizing image distortion in stego images. In this case, the edges of the cover images are calculated using an edge detector based on a CNN with Deep Supervision, which can preserve more edge pixels than the traditional edge detection methods. More bits should be inserted into edge pixels, and fewer bits should be inserted into non-edge pixels in this work's principal guideline for embedding secret bits within the cover image. The cover image was first pre-processed by masking the final five bits of each pixel. Next, a grayscale edge map was produced by applying the edge detector model. The grayscale edge map was transformed into a binary version utilizing both global and adaptive binarization algorithms to obtain dominant edge information. The Berkeley Segmentation Dataset and Benchmark (BSDS 500) [29] were used to train the models. Ten benchmark grayscale images from the USC-SIPI [30] image database were used for the experiment. Achieving an average SSIM of 0.960, average PSNR of 37.11 dB, and average embedding capacity of 3.45 bpp [18].

Zhu J et al. suggested a framework for data hiding based on generative adversarial networks (GANs) and autoencoder training. An adversarial discriminator, noise layer, decoder, and encoder make up the framework's four primary components. The initial encoder generated an encoded image that was visually identical to the original message, thereby allowing the decoder to reconstruct the original content. The inclusion of a noise layer adds distortions to the encoded image, resulting in a noisy version. The adversary, resembling the decoder in structure, performs binary classification. The system achieves an average PSNR of 37.68 dB and average SSIM of 0.985 on images from common objects in context (COCO) [31] database [19].

Ambika et al. introduced customized GAN steganography that was employed to address security concerns and reduce the loss of private information. This study is unique in that it extracted secret information from noise vectors using a user-supplied key, adding an extra security layer at both the encoding and decoding ends. An additional noise reduction layer was incorporated throughout the encoding and decoding stages to minimize the loss of confidential data. The system achieves an average PSNR of 38.96 dB with 5.61 bpp on images from the Pascal VOC [32] database [20].

Baluja proposed a neural network system concurrently trained to hide multiple images. The entire system consisted of three networks that were specifically designed to operate as a pair and were trained as a single network to produce hiding and revealing processes. The preparation network, which is the initial component, makes the image ready to be concealed. After receiving the output from the Preparation Network, the Hiding Network embeds the hidden image into the host image. The task of extracting the secret image from the container falls into the third component, the Reveal Network. The system achieved an average PSNR of 37.60 dB and an average SSIM of 0.970 on images from the ImageNet database [21].

Lu Shao-Ping et al. suggested a framework for multiple image hiding called an Invertible Steganography Network (ISN), which introduces the forward and backward propagation operations of a single invertible network to exploit the image embedding and extraction challenges and leverages steganography and the recovery of hidden images as a pair of inverse problems in image domain transformation. This work provides an increasing amount of research aimed at improving steganographic methods, particularly those based on neural network capabilities. The system achieved an average PSNR of 32.90 dB and average SSIM of 0.945 on images from the ImageNet database [22].

Das Abhishek et al. introduced a deep neural network designed for encoding and decoding multiple concealed images within a singular cover image of a matching resolution. The three networks that comprise the model are the prep, hidden, and reveal networks. The secret image was prepared using the prep network before being supplied as input to the hiding network, which created a container image using the output of the prep network and a cover image. Several prep-reveal networks constitute the proposed framework. The system achieved an average PSNR of 28.21 dB and an average SSIM of 0.826 on the images from the COCO database [23]. Guan Zhenyu et al. suggested a framework for multiple image hiding, called deep invertible image hiding (DeepMIH), which uses an invertible hiding neural network with an importance map module. The authors hope to solve the problem of hiding multiple images within a single cover image by fusing innovative neural network designs with the concept of invertibility. This study examines the trade-offs between visual appeal and concealment ability and offers insightful information about the state of several image-hiding strategies. The system achieved an average PSNR of 36.55 dB and an average SSIM of 0.961 on the images from the COCO database [24].

Chen Haoyu et al. suggested a framework for probabilistic multiple-image concealing. A pyramid of generative adversarial networks (SinGAN) was proposed to implicitly learn the patch distribution of a single cover image. The final step in disguising the image is distribution learning, in which we fit a deterministic mapping from a fixed set of noise maps to the secret image using the same SinGAN. A single forward propagation is sufficient for the recipient to extract a hidden image using the embedding key. The system achieved an average PSNR of 36.84 dB and an average SSIM of 0.958 on images from well-known standard images [25].

Li Fengyong et al. developed a deep invertible neural network with a spatial channel joint attention mechanism. A cascaded deep invertible neural network framework is built using a sequence of basic invertible networks with the same structure. This allows for the successive embedding of several confidential images into a single cover image via a set of adjustable cascaded iterative processes. The purpose of the spatial-channel joint attention module is to direct the embedding of sensitive data into the safer areas of the image. Which, owing to its large embedding capacity, can solve the issue of the visual quality of steganographic images. The system achieved an average PSNR of 37.48 dB and an average SSIM of 0.966 on images from the COCO database [26].

Ping Ping et al. introduced a new technique for hiding multiple images that uses reversible color transformation and up-sampling. The carrier image is upsampled via the interpolation technique, which enhances the effect of image concealing by utilizing the attributes of comparable neighboring pixel values in the upsampled image. Reversible color modification was then used in the embedding process to reduce the likelihood of local blurring in the stego-image. The system achieved an average PSNR of 33.61 dB and an average SSIM of 0.966 on images obtained from well-known standard images [27].

Table 1: Comparative Study On Multiple Image-Hiding.

Method	Dataset	Number of secret images	Number of samples	Technique	Results				Remarks	
			Training / Testing		Secret/ Recovery		Cover/ Stego		Pros	Cons
					SSIM	PSNR	SSIM	PSNR		
[21] (2020)	ImageNet	2	- 1000	Encoder and decoder networks	0.950	35.60	0.960	36.70	straightforward and simple architecture is used	unknown training size, dual hiding only
[22] (2021)	ImageNet and Paris Streetview	4	100,000 1000	Invertible Steganography Network	0.942	33.02	0.955	36.03	Concealing and revealing share same network parameters	Failure when using noisy cover images
[23] (2021)	Tiny ImageNet	3	2000 2000	Encoder and decoder networks	0.870	29.35	0.892	32.94	straightforward and simple architecture is used	Image size is 64 x 64 which is very small
[24] (2022)	DIV2K and COCO	4	100 5000	Invertible hiding neural network with an importance map module	0.918	32.98	0.943	34.90	Concealing and revealing share same network parameters	More hidden images worsen prior ones
[25] (2022)	DIV2K and COCO	4	- 400	A pyramid of generative adversarial networks	0.923	34.03	-	-	The Embedding key enhances the security of secret image reproduction.	Unknown training size, Cover /Stego pair results
[26] (2023)	DIV2K and COCO	3	100 5000	invertible neural network with a spatial channel joint attention	0.922	32.73	0.978	40.16	Concealing and revealing share same network parameters	High computation time
[27] (2023)	DIV2K and well-known standard images	3	- -	Reversible color transformation, up-sampling	-	30.83	-	32.96	High imperceptibility and security with low embedding capacity	Number of samples is not explained

3. METHODOLOGY

This section discusses a novel image-hiding architecture with the suggested techniques for interpolation, an invertible hidden neural network (IHNN), and super resolution (SR) to hide many images with high invisibility, high security, and large capacity. The general suggested process is described below, with subsections providing a thorough explanation of each step.

The pipeline of the proposed method is divided into two main phases. First, the forward concealing process, as shown in Figure. 2, takes a secret image(s) as the input and combines them using a forward interpolation module.

The cover image and combined secret image are then input to the forward pass of IHNN, which outputs the stego image. Second, the reverse revealing process, shown in Figure. 3, involves utilizing a stego image as the input, which is then passed through the backward pass of the IHNN. Subsequently, the IHNN outputs the combined secret and cover images. The combined secret image is then input backward, and the interpolation module outputs the recovered secret images, but in small sizes. The super-resolution module is used to recover secret images with the original size. It should be noted that revealing and concealing in the proposed image-hiding architecture are the opposite processes, sharing the same IHNN parameters.

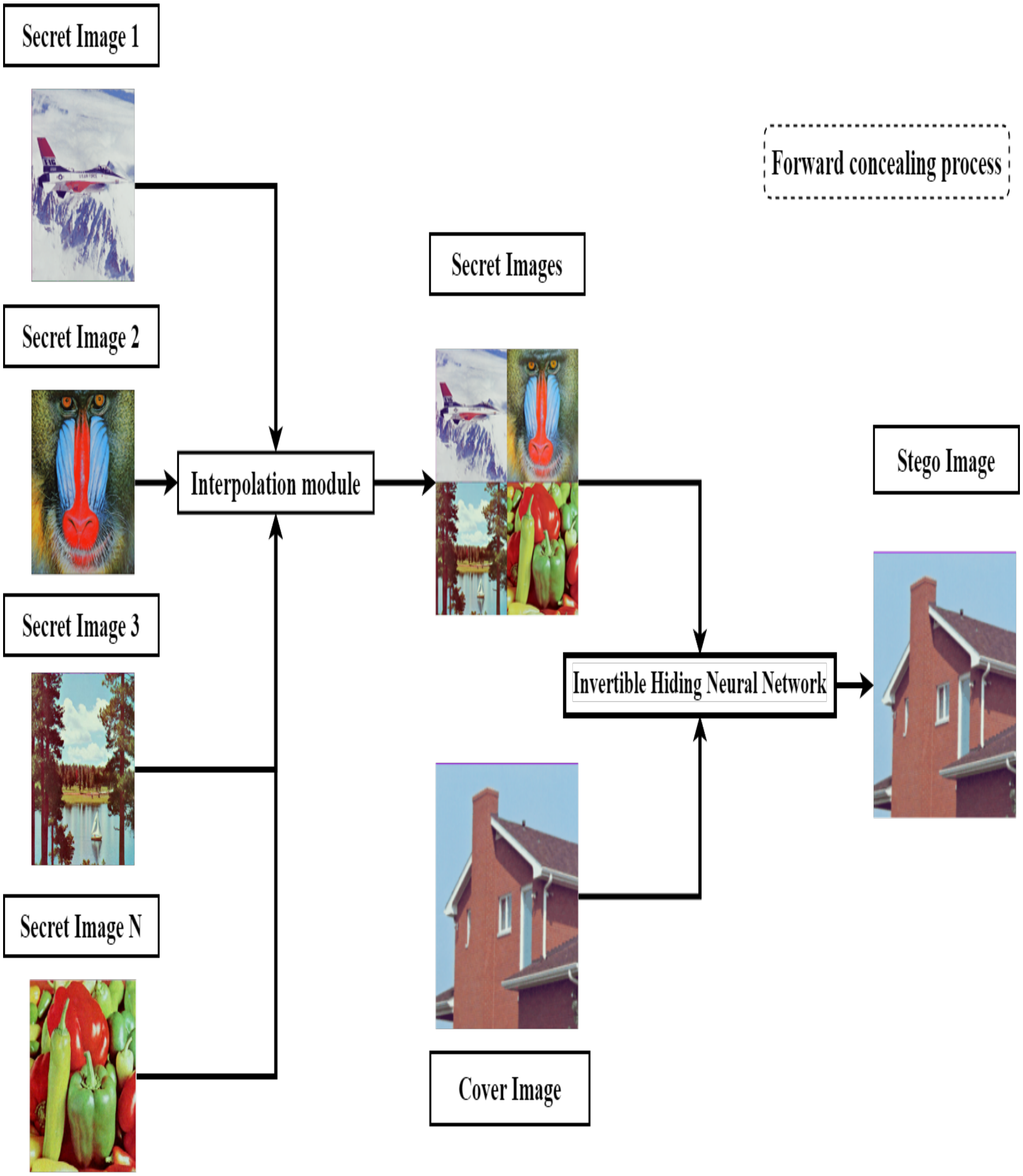


Figure 2: The forward concealing process with N = 4 secret images.

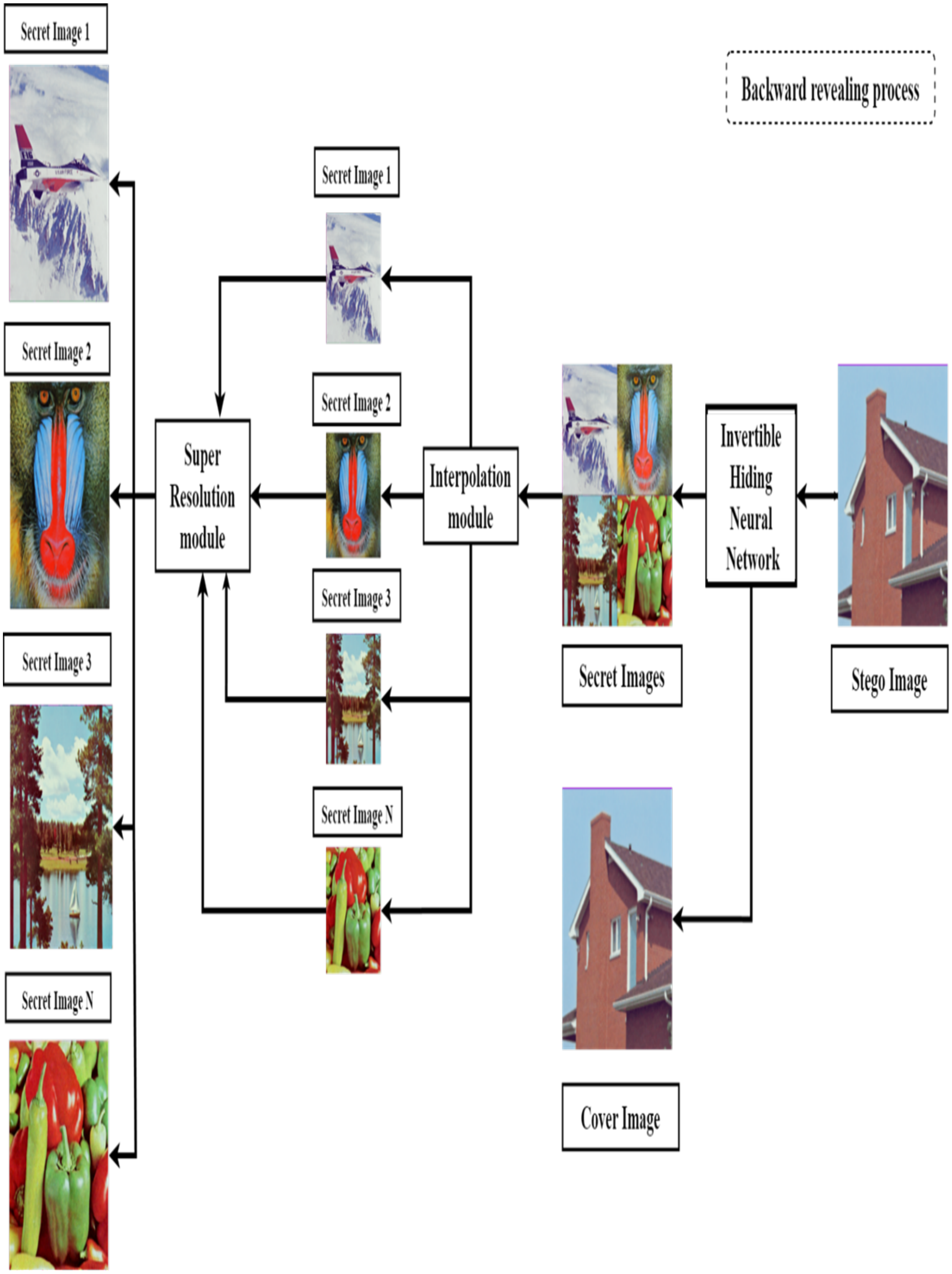


Figure 3: The backward revealing process with N = 4 secret images.

3.1 Preprocessing and Interpolation Module

The goal of this module is to implement an interpolation-based image-resizing technique that involves both forward and backward processes. There are several resized algorithms, such as nearest-neighbor interpolation, Lanczos interpolation, bilinear interpolation, and bicubic interpolation [34-36]. Bicubic interpolation is used because it reflects a balance between computational efficiency and preservation of visual quality, making it a preferred choice in many image-processing scenarios [36,37]. Note that the preprocessing step crops all images using the center-cropping strategy to ensure that the cover and all secret images have the same resolution.

In the forward process, N input images were resized according to the downscaling strategy, as shown in Table 2 and combined into a single larger image of the same size as the cover image. In the backward process, smaller-sized images are extracted from the combined image according to the downscaling factor in Table 2, using the same interpolation algorithm used in the forward process. Table 2 lists the scaling factor of the input secret images according to the range of the number of secret images to preserve the aspect ratio. Padding images of the same size as the secret images are added to reach the maximum size of each downscaling factor.

Table 2: Downscaling Factor Of The Input Secret Images.

scaling factor	Number of input secret images	
	Minimum	**Maximum**
2	2	4
4	5	16
8	17	64

3.2 Invertible Hiding Neural Network (IHNN)

As shown in Figure. 4, through a series of concealing blocks, a secret image is concealed in a cover image in the forward concealing process, producing a stego image together with lost information R. The secret image is recovered by a sequence of revealing blocks that receive the stego image and auxiliary variable Z from a Gaussian distribution in the backward revealing process. It should be noted that in the IHNN, revealing and concealing have the same network parameters because they are opposite processes.

Dinh et al [38]. Were the ones who initially introduced the invertible neural network (INN). When $x = f_\theta(y)$ is given and a variable y is to be recovered, y can be obtained directly as $y = f_\theta^{-1}(x)$, where the inverse function f_θ^{-1} is intended to have the same parameters (θ) as f_θ. They proposed multi-scale layers to lower the computational cost and improve the regularization ability of INN and convolutional layers in coupling models to improve INN's ability to handle image-related tasks [39]. Invertible 1 × 1 convolution and Glow were applied to INN by Kingma et al [40]., an effective method for realistic-looking image generation and manipulation, was proposed. Owing to its outstanding capabilities, INN has been applied to numerous image-related applications. Ouderaa et al [41]. Specifically used INN for an image-to-image translation task. Thus, to handle image concealing and revealing as invertible processes, Jing Junpeng et al [42]. Employed INN in the image hiding task as IHNN.

The IHNN's general framework is depicted in Figure. 4. A pair of cover image X_{cover} and secret image X_{secret} can be used as inputs in the forward concealing process. Discrete wavelet transform (DWT) is used to first break them down into low and high-frequency wavelet sub-bands, which are then fed into a series of concealing blocks. After passing through an inverse wavelet transform (IWT) block, the outputs of the final concealing block generate a stego image X_{stego} together with the lost information (R). To retrieve the secret image X_{secret}, the stego image X_{stego} and an auxiliary variable (Z) traverse the DWT and several revealing blocks in the backward revealing process. Table 3 shows the summary of notations used in this study.

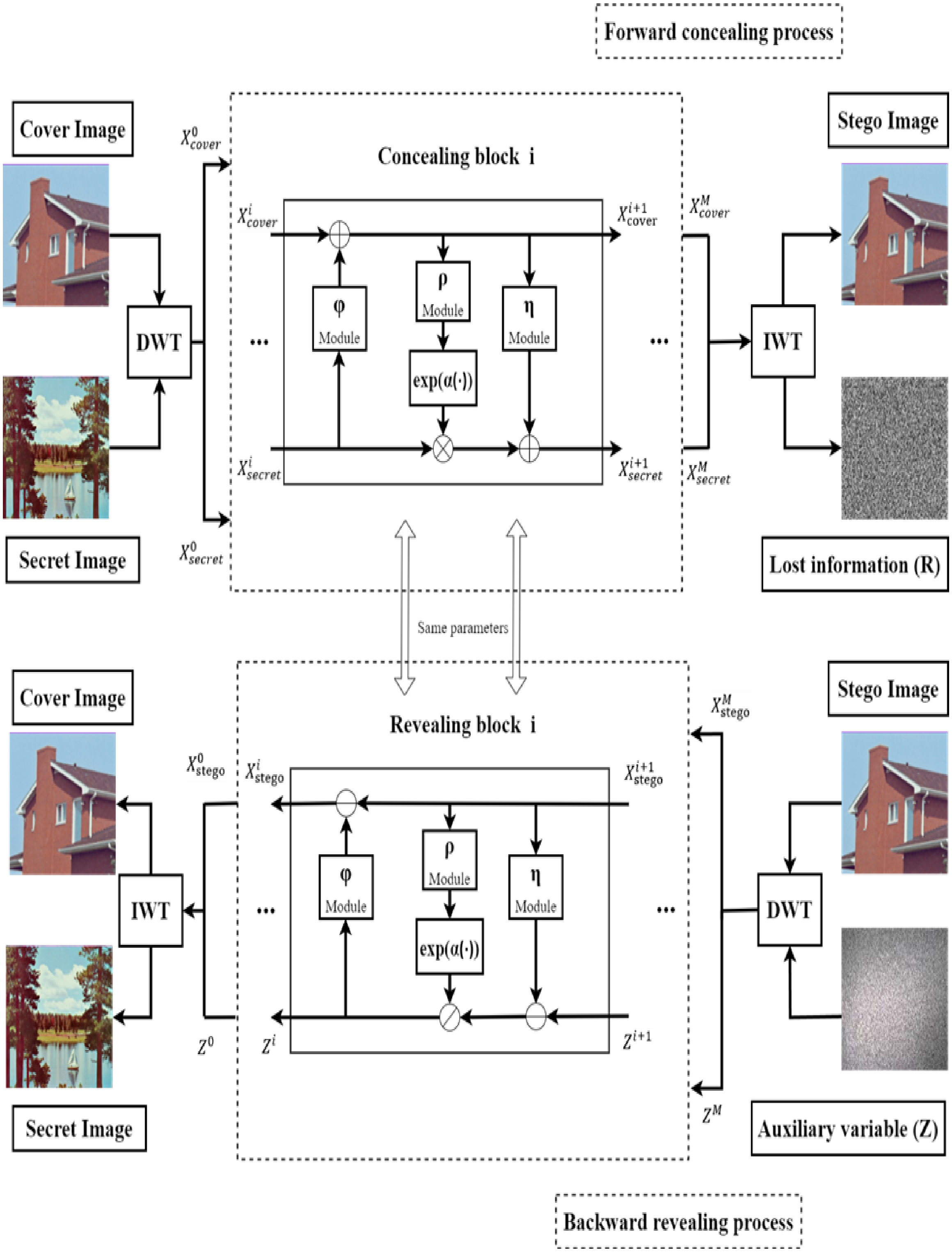


Figure 4: The IHNN framework.

Table 3: Summary Of Notations.

Notation	Description
$f(.)$	The forward process of IHNN
$f^{-1}(.)$	The backward process of IHNN
N	The number of secret images to be hidden
M	The number of invertible blocks
X_{secret}	The secret image to be hidden
X_{cover}	The cover image to hide secret image within
X_{stego}	The stego image with the secret image hidden within
$X_{recovery}$	The recovery secret image obtained from stego image
R	The lost information in the forward concealing process
Z	The auxiliary variable in the backward revealing process
B	The batch size of the feature map
C	The height of the feature map
H	The width of the feature map
W	The channel number of the feature map
$\alpha(\cdot)$	A sigmoid function multiplied by a constant factor
$\rho(\cdot)$, $\varphi(\cdot)$, and $\eta(\cdot)$	Arbitrary functions
I^{HR}	The original high-resolution image
I^{SR}	The reconstructed super-resolution image
Fm	The feature map extracted from the convolutional layer

DWT/IDWT Wavelet domain hiding: Color distortion and texture-copying artefacts are easily caused by image concealing in the pixel domain [43]. The frequency domain, particularly the high-frequency domain, is better suited for image concealing than the pixel domain. To improve the network's ability to fuse secret information into the cover image, we utilized DWT to split the image into low- and high-frequency wavelet sub-bands prior to entering the invertible blocks. Furthermore, the perfect reconstruction property of wavelets [44] can enhance the image-hiding performance and reduce information loss. The feature map with the dimensions (B, C, H, W) is transformed into (B, 4C, H/2, W/2) after the DWT. The DWT and IWT are conducted using the Haar wavelet kernel because of its efficiency and simplicity. Owing to its bidirectional symmetry, the wavelet transform does not have an impact on the end-to-end training of our network.

Invertible concealing and revealing blocks: As shown in Figure. 4, the revealing and concealing blocks share the same network parameters and submodules, but the information flow directions are different. The architecture, which is built as follows, Equation 1 is established by M concealing-revealing blocks with the same architecture. Equation (1) formula is used to formulate the outputs, X_{secret}^{i+1} and X_{cover}^{i+1} for the i-th concealing block in the forward process from the inputs X_{secret}^{i} and X_{cover}^{i}:

$$X_{cover}^{i+1} = \varphi\,(X_{secret}^{i}) + X_{cover}^{i}$$

$$X_{secret}^{i-1} = \eta(\,X_{cover}^{i+1}) + X_{secret}^{i} \cdot \exp\,(\alpha\,(\rho(\,X_{cover}^{i+1}))). \qquad (1)$$

The prevalent dense block is introduced in[45] due to its effective representation capabilities for arbitrary functions. Following the final concealing block, we acquire the outputs X_{secret}^{M+1} and X_{cover}^{M+1} . These outputs are subsequently input into two IWT blocks to produce the lost information R and the stego image X_{stego} . As shown in Figure. 4, the information flow direction during the revealing process is from the (i+1)-th revealing block to the i-th revealing block, which is the opposite order of the concealing process. In particular, the stego image X_{stego} and an auxiliary variable Z through DWT generate the inputs X_{stego}^{M+1} and Z^{M+1} for the M-th revealing block. Z is obtained randomly from a Gaussian distribution in this instance. X_{stego}^{M} and Z^{M} are the M-th revealing block's outputs. The inputs for the i-th revealing block are Z^{i+1} and X_{stego}^{i+1}, and the outputs are Z^{i} and X_{stego}^{i} as shown in Equation (2). The following is a model of their relationship:

$$Z^{i} = (Z^{i+1} - \eta(\,X_{stego}^{i+1}\,)) \cdot \exp\,(-\alpha\,(\rho(\,X_{stego}^{i+1}\,)))$$

$$X_{stego}^{i} = X_{stego}^{i+1} - \varphi(\,Z^{i}\,). \qquad (2)$$

The recovery image $X_{recovery}$ is produced by feeding the output X_{stego}^{1} into an IWT block following the final revealing block which is revealing block 1.

The lost information R and auxiliary variable Z: R represents the lost information generated during the forward concealing process, while Z serves as an input for the reverse revealing process besides the stego image. The network attempts to hide the secret image into the cover image throughout the concealing phase. Secret information is unavoidably lost because it is hard to conceal such a high capacity in the cover image. Furthermore, the original information in the cover image may be lost by the intrusion of the secret image. The lost secret information and destroyed cover information make up the lost information R. But since we don't have access to R during the backward revealing process, it's challenging to retrieve the initial hidden image. Z is an auxiliary variable that we introduced to replace R to solve this problem. More specifically, for the concealing and revealing processes to be invertible, R and Z must adhere to the same distribution. Since the following, it appears that R is case agnostic here. Assume that the dataset's overall image distribution is X. X_{cover} and X_{secret} are the inputs used in the forward process. They were generated from the same dataset and have the same distribution: X = X_{cover}, X_{secret}. The mixed distribution of the outputs X_{stego} and R should follow the same distribution as the inputs, that is, $X_{stego} \times R \sim X$, because of the accurate equivalence of Equations. (1) and (2) and the reversible requirement of INN. The concealing loss for the stego image X_{stego} is pushed to match the cover image's distribution, $X_{stego} \sim X$, in the subsequent Loss function section. It is plausible to presume that the remaining R is case agnostic as a result. Without having access to R, the recovery image $X_{recovery}$ must be retrieved from just the stego image X_{stego} in backward revealing. Z, the variable, is drawn at random from a distribution that is a case agnostic distribution. Every sample in the distribution is guaranteed to be able to recover the secret information since the distribution is learned during training through the revealing loss in the subsequent Loss function section. Here, we assume the distribution to be Gaussian, i.e., $Z \sim N(\mu_0, \sigma_0^2)$, without losing generality.

Loss function: The three losses that make up the total loss function are the low-frequency wavelet loss (L_{freq}) to improve the hiding security and quality, the revealing loss (L_{rev}) to ensure the recovering performance, and the concealing loss (L_{con}) to assure the concealing performance[42]. We then go into more detail about these losses.

Concealing loss. The goal of the forward concealing process is to create a stego image called X_{stego} by hiding X_{secret} into X_{cover}. It is necessary for the stego image to be inconspicuous with the cover image. In pursuit of this objective,

$$L_{con}(\theta) = \sum_{n=1}^{N} l_C\left(X_{cover}^{(n)}, X_{stego}^{(n)}\right), \tag{3}$$

where f_θ ($X_{cover}^{(n)}$, $X_{secret}^{(n)}$) is the value of $X_{stego}^{(n)}$, and θ denotes the network parameters. Furthermore, N denotes the quantity of training samples, and l_C, which can be either 1 or 2 norm, quantifies the difference between the cover and stego images as shown in Equation (3).

Revealing loss: The network should be able to reconstruct the secret image in the backward revealing process using any sample of Z from the Gaussian distribution p(z), given the stego image X_{stego} produced from the forward concealing process. To accomplish this, we define L_{rev} as,

$$L_{rev}(\theta) = \sum_{n-1}^{N} \mathbb{E}_{z\sim p(z)}\left[l_R\left(X_{secret}^{(n)}, X_{recovery}^{(n)}\right)\right]. \tag{4}$$

where f_θ^{-1} ($X_{stego}^{(n)}$, Z), is the recovery image $X_{recovery}^{(n)}$, along with $f_\theta^{-1}(\cdot)$ is the backward revealing process. Like l_C, l_R evaluates the difference between ground-truth secret images X_{secret} and recovered secret images $X_{recovery}$ as shown in Equation (4).

Low-frequency wavelet loss: To improve the network's resistance to steganalysis and improve the concealing quality, a low-frequency wavelet loss (L_{freq}) was added [17,42], in addition to the first two losses. This loss is motivated by [17,42], which indicates that information concealed in high-frequency components has a lower detection probability than information concealed in low-frequency components. Here, wavelet decomposition of the low frequency sub-bands of the stego image must resemble that of the cover image to guarantee that much of the information is concealed in the high frequency sub-bands. Assuming that $H(\cdot)_{LL}$ represents the process of obtaining low-frequency sub bands following wavelet decomposition, the following Equation (5) formula applies to L_{freq},

$$L_{freq}(\theta) = \sum_{n=1}^{N} l_F \left(H(X_{cover}^{(n)})_{LL}, H(X_{stego}^{(n)})_{LL} \right). \quad (5)$$

The difference between the low-frequency sub-bands of the cover and stego images is measured here by l_F.

Total loss function: As shown in Equation (6), the overall loss function (L_{total}) is the weighted total of the low-frequency wavelet loss, the revealed loss L_{rev}, and the concealing loss L_{con}. L_{freq} in this manner,

$$L_{total} = \lambda_c L_{con} + \lambda_r L_{rev} + \lambda_f L_{freq}. \quad (6)$$

In this case, the weights λ_c, λ_r, and λ_f are used to balance various loss terms. First, we pre-train the network throughout the training process by setting λ_f to zero and minimizing L_{con} and L_{rev}. The network is then trained end-to-end by adding L_{freq}.

3.3 Super Resolution Module

One of the most important image-processing techniques for enhancing image resolution in computer vision is image super-resolution (SR), which is the process for generating high-resolution (HR) images from equivalent low-resolution (LR) images. Super-resolution techniques can be broadly divided into two primary categories: deep-learning techniques and traditional techniques. Single-image super-resolution (SISR) and multiple-image approaches are the two categories to which the current image super-resolution techniques belong. While learning is performed for many LR-HR pairs for a given scene in multiple-image SR, it is performed for a single LR-HR pair for a single image in a single-image SR [45-49].

For the proposed task as shown in Figure. 5, Real Enhanced Super-Resolution Generative Adversarial Networks (Real-ESRGAN) is employed [46], a deep learning technique for single image super resolution. Fig 5 shows that Real-ESRGAN architecture generates HR image from a corresponding LR image. A deep network with multiple residual-in-residual dense blocks (RRDB) is used, following the same SR network in ESRGAN [49]. Additionally, pixel-unshuffle was used to increase the channel size, decrease the spatial size, and achieve super-resolution with a scale factor of ×2 on the original ×4 ESRGAN architecture. The Real-ESRGAN is simple to train and efficient in generating HR images from LR ones with minimal artifacts.

Loss function: For many real-world images, the Real-ESRGAN, trained using synthetic data and a combination of L1 loss, perceptual loss[50], and adversarial/GAN loss [49]. The details of the three losses that constitute the total loss function are discussed below.

L1 loss: The L1 loss function is also known as mean absolute error (MAE). Equation (7) shows how this loss function calculates the average absolute difference between HR and LR images. The model learns to generate HR images that closely approximate the LR one, maintaining fine details and sharp edges by minimizing the L1 loss during training. It helps eliminate over-smoothing in the output images and is more resilient to outliers than other loss functions, such as L2 loss. Therefore, the L1 loss is chosen for optimizing our SR network.

$$L_1 = \frac{1}{WHC} \sum_{x,y,c} \left| I^{SR}{}_{x,y,c} - I^{HR}{}_{x,y,c} \right| \quad (7)$$

Perceptual loss: The main goal of perceptual loss is to maximize the perceptual similarity between the SR and LR images by optimizing the network in the content (feature) space instead of the image (pixel) space, which concentrates on maintaining the original image's content while creating a new one. Finding the content features (high-level features) for the original LR image and the resultant HR image is the first step in defining the perceptual loss. Perceptual loss is calculated as the mean square error between these extracted features, as shown in Equation (8).

$$l_{perceptual} = \frac{1}{W_i H_i C_i} \sum_{x,y,c} \left(\mathrm{Fm}_i(I^{SR})_{x,y,c} - \mathrm{Fm}_i(I^{HR})_{x,y,c} \right)^2 \quad (8)$$

Adversarial/GAN loss: The generator was designed to fool the discriminator because the adversarial/GAN loss purpose is to fool the discriminator. As shown in Equation (9), where θ_D represents the discriminator network's parameters and $D_{\theta_D}(I^{SR})$ is the probability that the generated HR image will resemble a realistic LR image from the discriminator's perspective. The adversarial loss function helps to optimize the generator learning process by penalizing the generator if it determines that the generated image is not realistic with a certain probability.

$$l_{adv} = -\log\left(D_{\theta_D}(I^{SR}) \right) \quad (9)$$

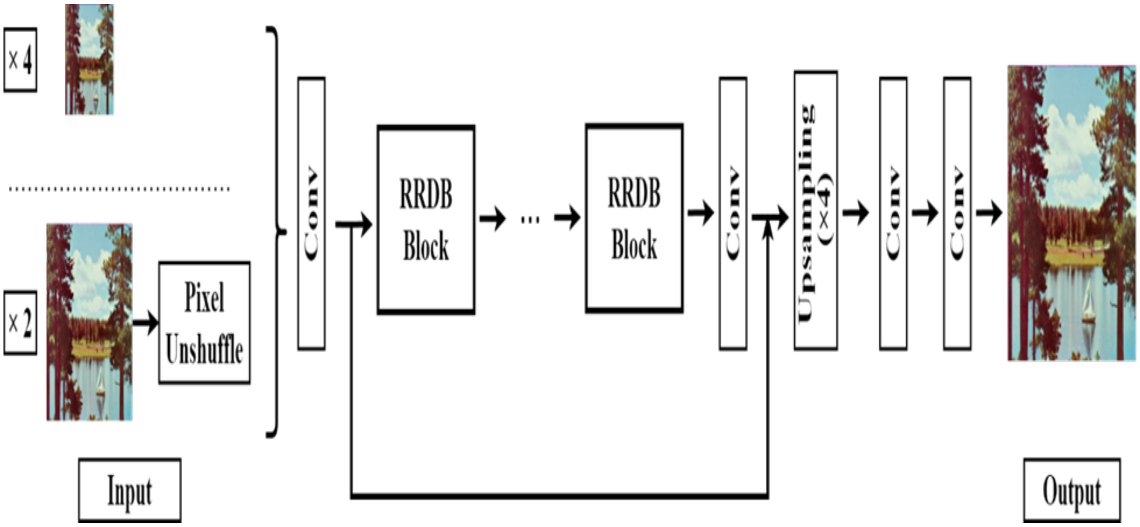


Figure 5: Real-ESRGAN architecture.

Total loss function: Equation (10) shows that the weighted sum of L1 loss, perceptual loss, and adversarial/GAN loss produces the overall loss function (L_{total}). The weighting parameters, denoted as λ ′ s determine the contribution of each loss term in maintaining the balance of the total loss function.

$$L_{total} = \lambda_1 L_1 + \lambda_2 L_{perceptual} + \lambda_3 L_{adv}. \qquad (10)$$

4. EXPERIMENTAL RESULTS

This section provides a detailed description of the experimental settings, including the datasets, benchmarks, and evaluation metrics used. In addition, the results for hiding dual secret images (N = 2) are presented. In addition, the results of hiding multiple secret images (N > 2) are presented. All the experiments were performed on a computer with a single NVIDIA RTX 3060 Ti GPU.

4.1 Experimental Settings

Datasets and settings: The IHNN was trained using the DIV2K [33] training dataset. The testing datasets were as follows: ImageNet [28] had 50,000 images at a resolution of 256 × 256, COCO [31] had 5,000 images at the 256 × 256 and DIV2K [33] had 100 images at a resolution of 1024 × 1024. There are sixteen concealing and revealing blocks in block M. It should be noted that the center-cropping technique was used to crop the test images to ensure that the resolution of the cover and secret images was the same. There was a total of 80 K iterations with a 256 × 256 training patch size. The values of λ_c, λ_r, and λ_f were set as 10.0, 1.0, and 10.0, respectively. With a mini-batch size of 16, half of the patches were secret patches, and the other half were chosen at random to be cover patches. Standard settings and a starting learning rate of 1 × $10^{-4.5}$, which is halved every 10 K iterations, were applied to the Adam optimizer [51].

Benchmarks: To confirm the efficiency of the proposed approach in concealing multiple secret images, we compare it with multiple state-of-the-art (SOTA) techniques for image hiding. This study follows an experimental research design, where the proposed method is implemented and evaluated against existing techniques using quantitative and qualitative image quality metrics. These techniques comprise a traditional image steganography method called k-LSB [52] with k=4, which conceals four bits, as well as five deep-learning-based approaches: HiDDeN [19], Baluja [21], Weng et al. [53], DeepMIH [24], and iSCMIS [26]. We used the same training dataset as ours to retrain the HiDDeN [19], Baluja [21], and Weng et al. [53] models to ensure a fair comparison. It should be noted that the image-hiding configuration used in this work is inconsistent with the original HiDDeN model, which could only hide messages. We changed its output size slightly to enable it to conceal images and then retrained the network.

Evaluation metrics: The Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) were the four metrics used to assess the quality of cover/stego and secret/recovery pairings [54-56]. These values were calculated as follows:

SSIM: The Structural Similarity Index (SSIM), a perception-based tool, evaluates image degradation as observed alterations in the structural information. It considers the variance and covariance measurements of an image's dynamic pixel range. Unlike MSE and PSNR, which evaluate absolute inaccuracy, SSIM considers the dependency between geographically nearby pixels by combining the brightness and contrast masking. The SSIM is given in the interval [-1.0,1.0], where 1 signifies two identical images. The SSIM computation formula for the two images, X(cover) and Y(stego), is as shown in Equation (11). The variances of the images are denoted by σx^2 and σy^2 and the mean by μx and μy.

$$SSIM\ (X, Y) = \frac{(2\mu x\ \mu y + 0.01)(2\sigma x\sigma y + 0.03)}{(\mu x^2+\mu y^2+0.01)+(\sigma x^2+\sigma y^2+0.03)} \quad (11)$$

PSNR: Peak signal to noise ratio (PSNR) is used to evaluate the performance, imperceptibility, and invisibility of the proposed steganography technology. The highest-quality representation of the cover image divided by the stego image is known as the PSNR. Decibels (dB) are used to indicate the quality of the reconstructed stego image, and a high PSNR score indicates that the image quality is suitable. As shown in Equation (12), the computation formula of Mean Squared Error (MSE) for two images with dimensions W × H, X(cover), and Y(stego), which defines PSNR.

$$MSE(X,Y) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(X_{i,j} - Y_{i,j})^2 \quad (12)$$

As shown in Equation (13), the PSNR calculation formula for two images X(cover) and Y(stego).

$$PSNR\ (X, Y) = 10log_{10}\left(\frac{255^2}{MSE(X,Y)}\right) \quad (13)$$

MAE: Mean Absolute Error (MAE) computes the difference(error) between two images which is like MSE, but it does so by taking the absolute value of the difference between images, which is shown in Equation (14):

$$MAE\ (X,Y) = \frac{1}{WH}\ \Sigma_{i=1}^{W}\Sigma_{j=1}^{H}\ |X_{i,j} - Y_{i,j}| \quad (14)$$

RMSE: Root Mean Square Error (RMSE) measures the difference(error) between two images. Based on the definition of MSE in Eq. (12), the RMSE is shown in Equation (15):

$$\text{RMSE} = \sqrt{MSE} \quad (15)$$

Higher image quality is indicated by larger values for SSIM and PSNR and smaller values for MAE and RMSE.

4.2 Results for Hiding Dual Secret Images(N = 2)

To implement dual secret image hiding, the super-resolution module (Real-ESRGAN) [46] is used with an upscaling factor of two combined with the IHNN. Tables 4 and 5 show the numerical results of the IHNN compared with the other comparison methods on DIV2K, ImageNet and COCO datasets in terms of SSIM, PSNR, RMSE and MAE with bold best results and underlying Second-Best Results. These two tables show that for the cover/stego image pair, our IHNN performs much better than the other methods in all four metrics. However, DeepMIH [24] performs better for the secret/recovery image pair because of the importance map module, which is intended to direct the present hiding procedure based on the concealing outcomes of prior operations. The traditional K-LSB approach [52] has low recovery quality and is unable to achieve dual image hiding because of its limited concealing capacity (2 bits for the first secret image and 2 bits for the second secret image).

Tables 4 and 5 demonstrate the good performance of the proposed model on the ImageNet and COCO datasets, even though it was trained only on the DIV2K dataset. This indicates how well the proposed model generalizes, which is crucial for real-world applications.

4.3 Results for Hiding Multiple Secret Images (N > 2)

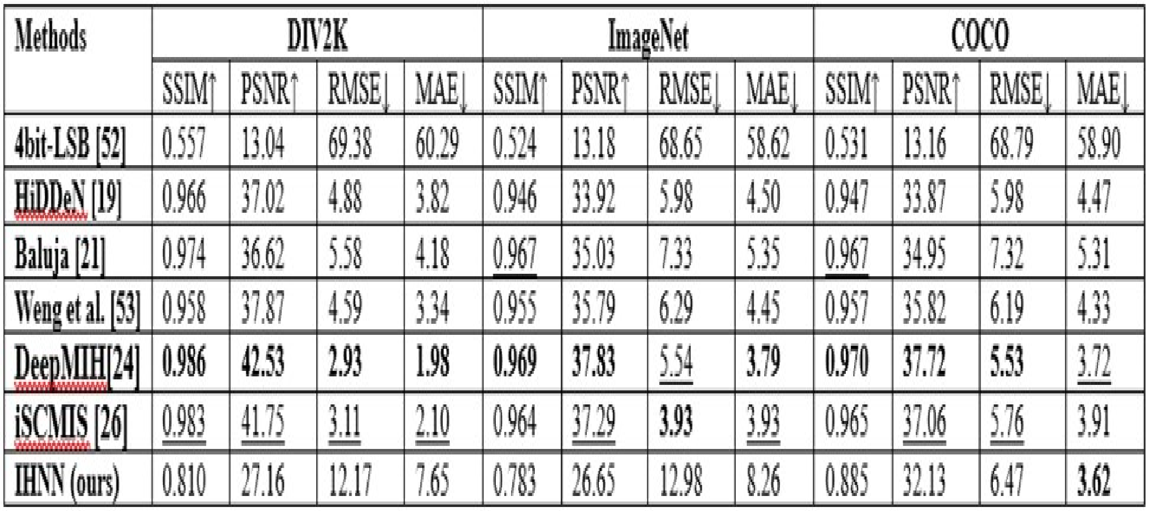
The super-resolution module (Real-ESRGAN) [46] is employed in conjunction with IHNN to hide various numbers of secret images into a single cover image to investigate the concealing capability of our proposed framework.

Quantitative Results: Table 6 lists the outcomes of hiding varying numbers of secret images on the DIV2K, ImageNet, and COCO datasets with two, four, and eight upscaling factors, respectively, in terms of SSIM, PSNR, RMSE, and MAE.

This table shows that the overall hiding performance deteriorates as more secret images are concealed. For example, the average PSNR value of secret images decreases by approximately 3.00 dB as the number of hidden images increases.

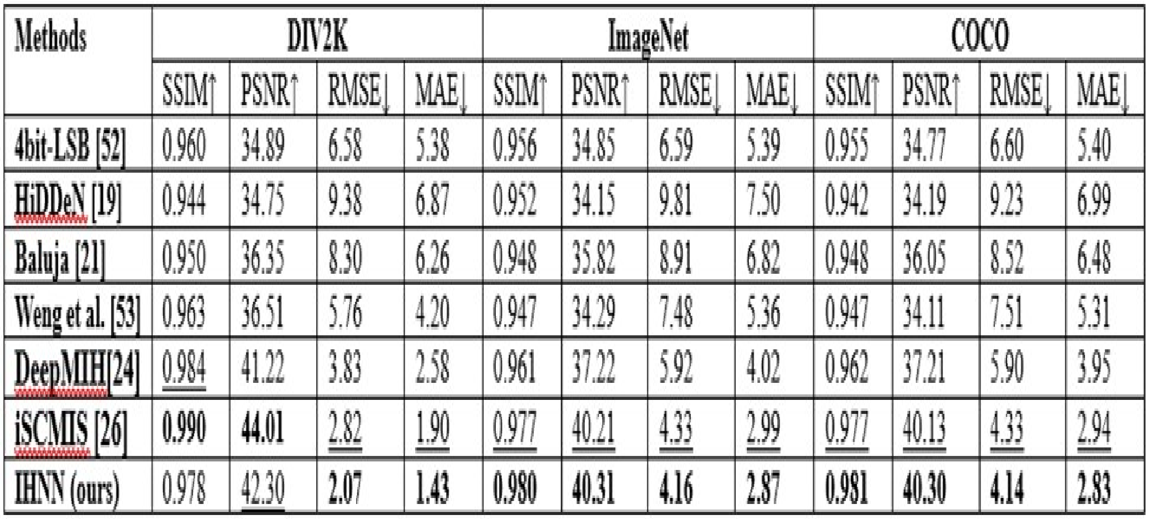
The experimental results confirm that the proposed method achieves the stated objectives. Specifically, the framework successfully concealed up to 64 images with an average PSNR of 20.51 dB and SSIM of 0.527, demonstrating improved capacity and imperceptibility compared to existing methods.

Table 4: Comparing Benchmarks Across Various Datasets for the Secret/Recovery Image Pair.



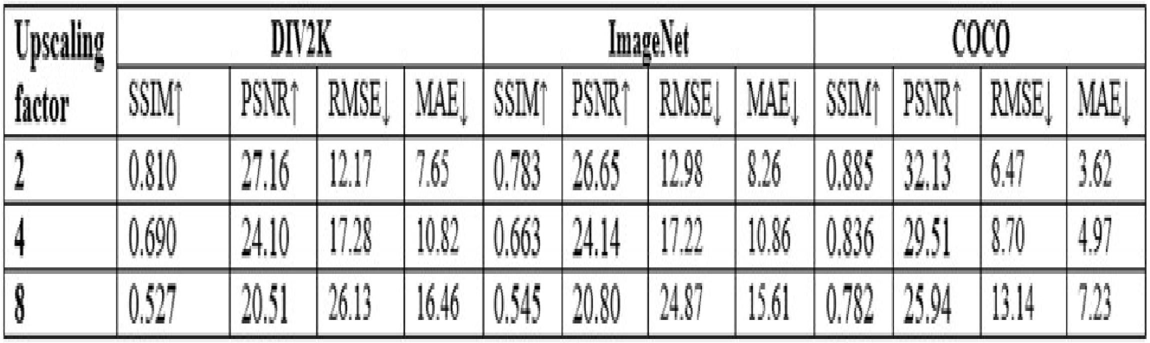
Methods	DIV2K				ImageNet				COCO			
	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓
4bit-LSB [52]	0.557	13.04	69.38	60.29	0.524	13.18	68.65	58.62	0.531	13.16	68.79	58.90
HiDDeN [19]	0.966	37.02	4.88	3.82	0.946	33.92	5.98	4.50	0.947	33.87	5.98	4.47
Baluja [21]	0.974	36.62	5.58	4.18	0.967	35.03	7.33	5.35	0.967	34.95	7.32	5.31
Weng et al. [53]	0.958	37.87	4.59	3.34	0.955	35.79	6.29	4.45	0.957	35.82	6.19	4.33
DeepMIH[24]	**0.986**	**42.53**	**2.93**	**1.98**	**0.969**	**37.83**	5.54	**3.79**	**0.970**	**37.72**	**5.53**	3.72
iSCMIS [26]	0.983	41.75	3.11	2.10	0.964	37.29	**3.93**	3.93	0.965	37.06	5.76	3.91
IHNN (ours)	0.810	27.16	12.17	7.65	0.783	26.65	12.98	8.26	0.885	32.13	6.47	**3.62**

Table 5: Comparing Benchmarks Across Various Datasets for the Cover/Stego Image Pair.



Methods	DIV2K				ImageNet				COCO			
	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓
4bit-LSB [52]	0.960	34.89	6.58	5.38	0.956	34.85	6.59	5.39	0.955	34.77	6.60	5.40
HiDDeN [19]	0.944	34.75	9.38	6.87	0.952	34.15	9.81	7.50	0.942	34.19	9.23	6.99
Baluja [21]	0.950	36.35	8.30	6.26	0.948	35.82	8.91	6.82	0.948	36.05	8.52	6.48
Weng et al. [53]	0.963	36.51	5.76	4.20	0.947	34.29	7.48	5.36	0.947	34.11	7.51	5.31
DeepMIH[24]	0.984	41.22	3.83	2.58	0.961	37.22	5.92	4.02	0.962	37.21	5.90	3.95
iSCMIS [26]	**0.990**	**44.01**	2.82	1.90	0.977	40.21	4.33	2.99	0.977	40.13	4.33	2.94
IHNN (ours)	0.978	42.30	**2.07**	**1.43**	**0.980**	**40.31**	**4.16**	**2.87**	**0.981**	**40.30**	**4.14**	**2.83**

Table 6: Comparison of various datasets for hiding different numbers of the Secret/Recovery Image Pair.



Upscaling factor	DIV2K				ImageNet				COCO			
	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓	SSIM↑	PSNR↑	RMSE↓	MAE↓
2	0.810	27.16	12.17	7.65	0.783	26.65	12.98	8.26	0.885	32.13	6.47	3.62
4	0.690	24.10	17.28	10.82	0.663	24.14	17.22	10.86	0.836	29.51	8.70	4.97
8	0.527	20.51	26.13	16.46	0.545	20.80	24.87	15.61	0.782	25.94	13.14	7.23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

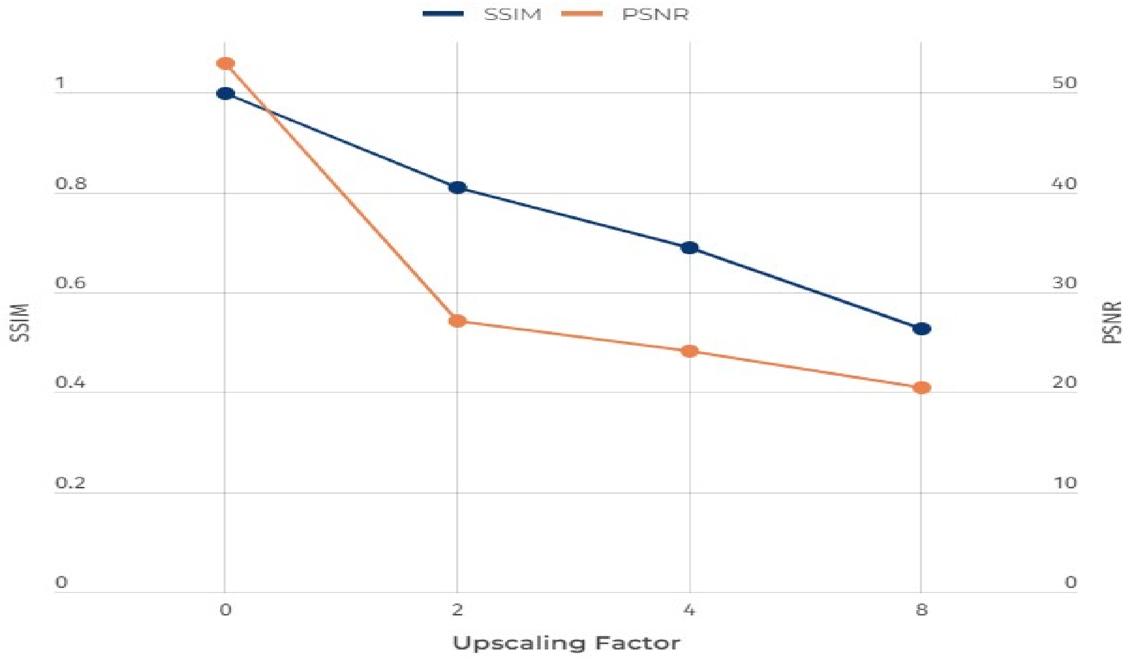


Figure 6: The relationship between the SSIM and PSNR with upscaling factors for the DIV2K dataset.

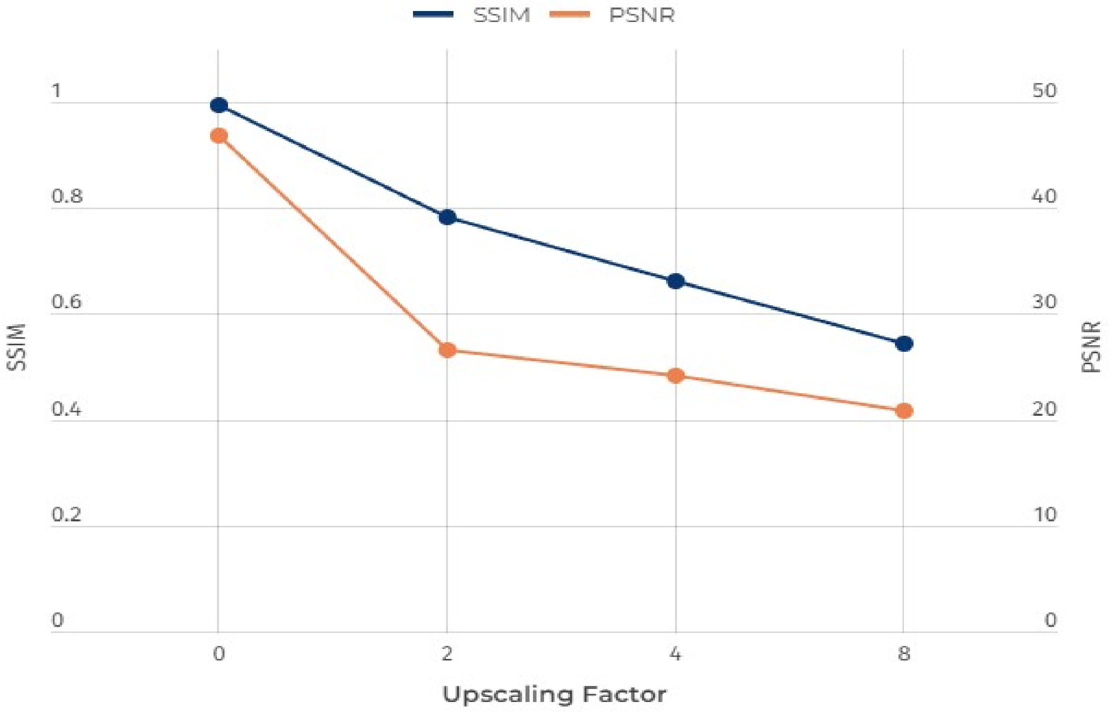


Figure 7: The relationship between the SSIM and PSNR with upscaling factors for the ImageNet dataset.

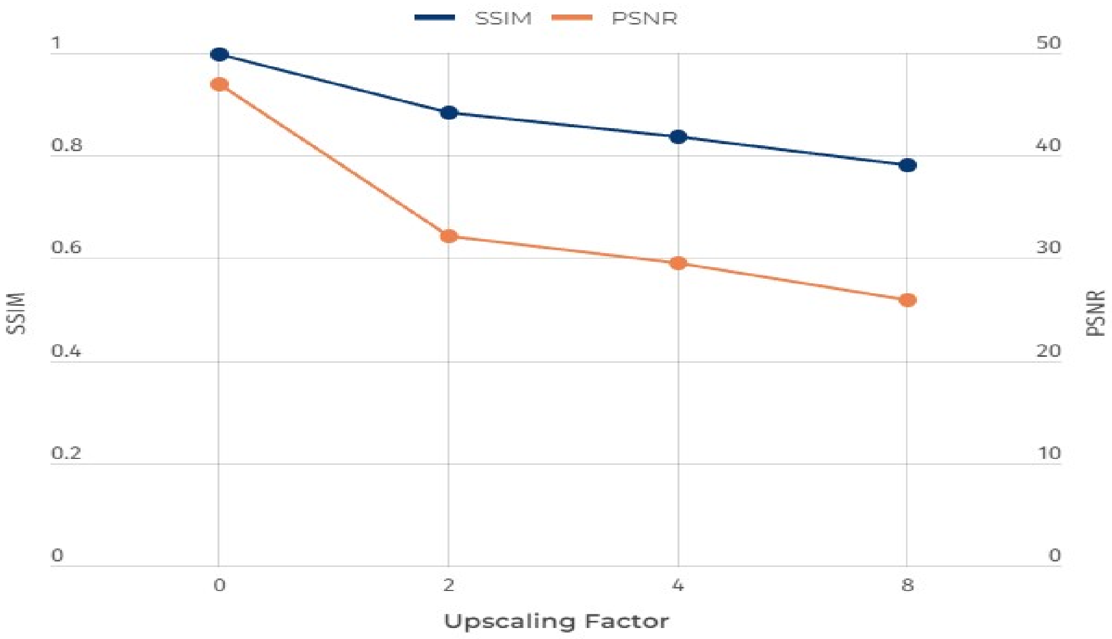


Figure 8: The relationship between the SSIM and PSNR with upscaling factors for the COCO dataset

.Table 7 compares the performance of DeepMIH [24], iSCMIS [26] and IHNN (ours) in hiding multiple secret images, as measured by the SSIM and PSNR values over the COCO dataset. DeepMIH demonstrated a relatively high SSIM and PSNR across different numbers of hidden secret images. iSCMIS provides comparable results but lacks data for the four hidden secret images. IHNN shows competitive performance in maintaining image quality while concealing multiple secret images, indicating its effectiveness in hiding multiple secret images.

Qualitative Results: A qualitative comparison of each upscaling factor is shown in Figure. 9. As shown in this figure, upscaling factors 2 and 4 allow for preservation of complex details cannot be accurately restored with an upscaling factor of eight, even though the visual quality of the hidden images was not significantly affected.

Table 7: Comparison of our IHNN for hiding three/four secret images (S = 3 and 4) over COCO dataset for the Secret/Recovery Image Pair.

Amount	**DeepMIH [24]**		**iSCMIS [26]**		**IHNN (ours)**	
	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑
3	0.930	33.87	0.922	32.73	0.885	32.13
4	0.917	32.98	-	-	0.885	32.13

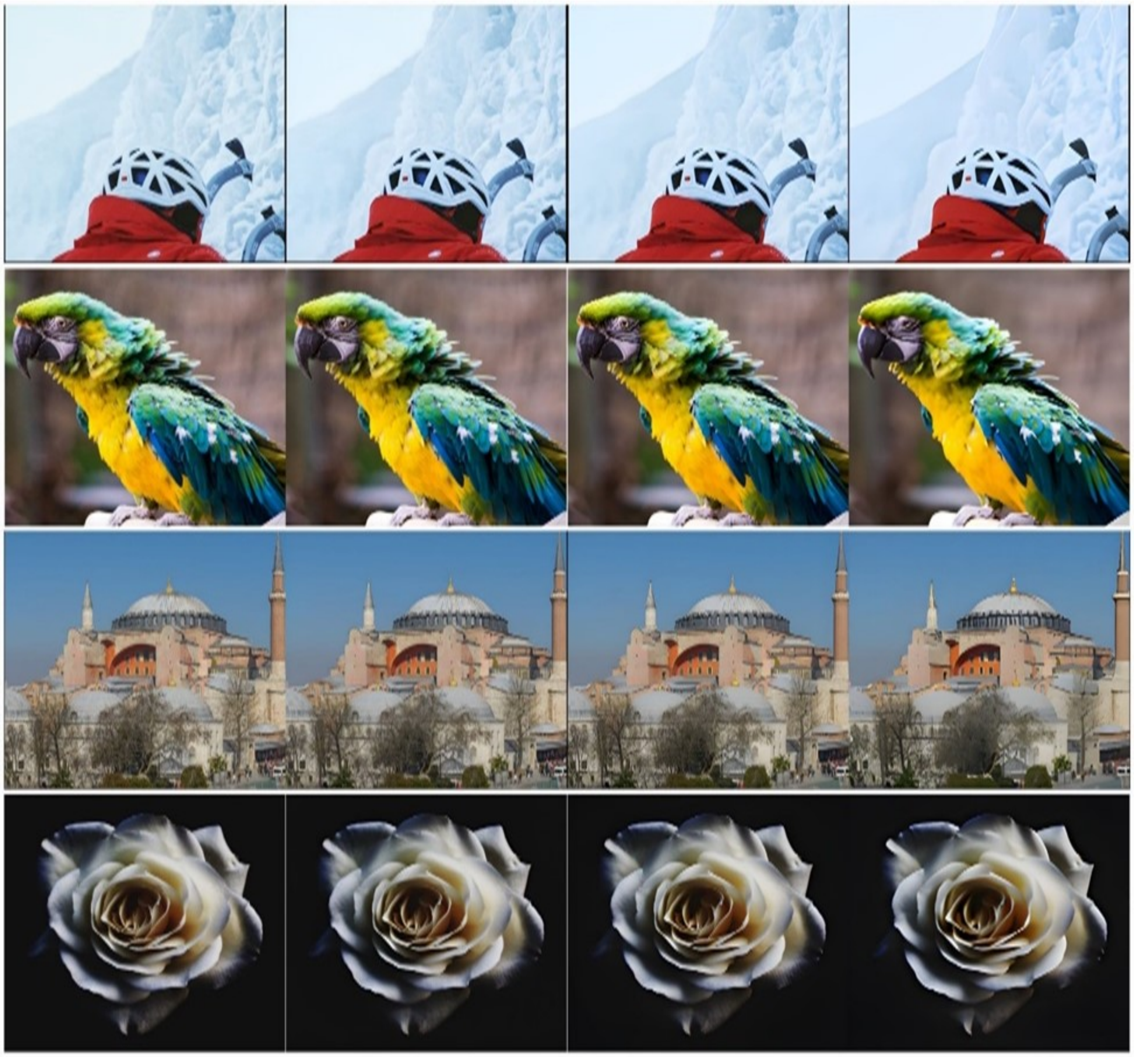
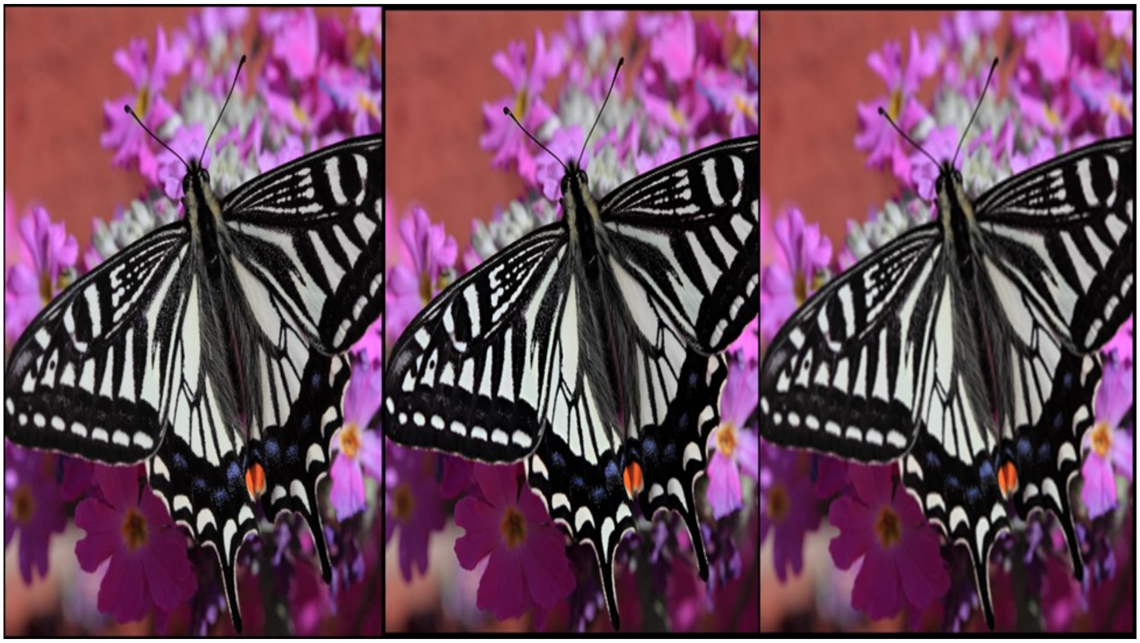


Figure 9: A comparison between the original and recovered secret images for each upscaling factor. ***(Zoom in for the best view)***

4.4 Ablation Study

Effectiveness of super-resolution module. Maximizing PSNR and minimizing MSE are typical metrics used to assess and contrast SR techniques [45-48]. MSE and PSNR are based on pixel-by-pixel image differences and their capacity to capture perceptually meaningful variations, including high texture detail, which is severely constrained [49]. The highest PSNR did not always correspond to a perceptually better SR result, as shown in Figure. 10.

Considering the significance of human perception of images, we conducted a more recent experiment to confirm the effectiveness of the proposed methodology.



Original Real-ESRGAN (24.13,0.783) Bicubic (27.48,0.870)

Figure 10: From left to right: original HR image, Real-ESRGAN, and bicubic interpolation. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling] ***(Zoom in for best view)***

Subjective Results: A Mean Opinion Score (MOS) test [49] was conducted to measure how well our approach recreated visually compelling content. We specifically asked 200 raters, to give the super-resolved images an integral score ranging from 1 (poor quality) to 5 (great quality). A higher MOS value suggests that the secret images are more hidden. Three versions of images bicubic, Real-ESRGAN [46], and the original HR image—were scored by the raters from the DIV2K, ImageNet, and COCO datasets. Thus, 84 instances—2x upscaling factor versions of 4 images, 4x version of 16 images, and 8x version of 64 images—that were presented in a randomized manner were scored by each rater. We discovered strong dependability and no discernible variations in evaluations of the same images. Table 8 provides a summary of the experimental findings from the MOS tests.

Table 8: Comparison of MOS testing values between bicubic interpolation and Real-ESRGAN on Various Datasets.

Upscaling factor	Bicubic	Real-ESRGAN
2	4.137	4.587
4	3.044	3.828
8	2.437	3.644
Average	**3.206**	**4.020**

Table 8 demonstrates that Real-ESRGAN [46] obtained better results with all upscaling factors and an average of approximately 0.8 better than that of the traditional method. The subjective results shown in Table 8 validate that our framework outperforms the bicubic method and proves the effectiveness of concealing many secret images.

5. CONCLUSION AND FUTURE WORK

In this study, A novel approach was developed based on the IHNN and super-resolution methods for hiding multiple secret images. To achieve this, full architecture was introduced, which is divided into two main phases: Forward concealing process and backward revealing process. In the forward concealing process, the secret images were first combined, and the cover image was then inputted into the IHNN, which outputs the stego image. In the backward revealing process, the stego image is fed as input to the IHNN, which outputs the combined secret image and cover image. The combined secret image is then separated to output the recovered secret images, which are small. Using the Super Resolution module to recover secret images with the original size. These methods significantly enhance the capacity to hide several images. Comprehensive experimental findings, both quantitative and qualitative, as well as subjective, demonstrate that the proposed method can achieve high invisibility and multiple image concealment with a large capacity. Unlike state-of-the-art methods, which struggle with a trade-off between hiding capacity and imperceptibility, our method successfully conceals up to 64 secret images while maintaining a high level of reconstruction accuracy. The combination of IHNN and super-resolution is novel in this context, enhancing image quality and ensuring more effective secret image retrieval.

There are still some limitations even though the proposed scheme shows high-capacity multi-image concealing with good imperceptibility and recovery accuracy. Real-time applications are challenging due to the model's comparatively high computing cost. Furthermore, it is necessary to conduct additional research because the method's efficacy may differ based on the cover image's texture and contrast.

In the future, we will consider experimenting with the proposed approach on other widely used datasets. More experiments should be conducted to ensure that the proposed framework ensures lossless reconstruction, enhances details, and improves the quality of the secret images. It is possible to expand the research on a group of traditional and deep-learning image steganography techniques to benefit from a combination of approaches.

REFERENCES

[1] Subramanian, N., Elharrouss, O., Somaya, A. M. & Bouridane, A (2021) Image steganography: A review of the recent advances. IEEE Access.

[2] Himthani, Varsha, Vijaypal Singh Dhaka, Manjit Kaur, Geeta Rani, Meet Oza, and Heung-No Lee (2022) Comparative performance assessment of deep learning-based image steganography techniques. Scientific Reports 12(1). https://doi.org/10.1038/s41598-022-17362-1

[3] Shi Dong, Ping Wang, Khushnood Abbas (2021) A survey on deep learning and its applications. Computer Science Review 40, p. 100379.

[4] Aiman Jan, Shabir A. Parah, Bilal A. Malik, Mamoon Rashid (2021) Secure data transmission in IoTs based on CLoG edge detection. Future Generation Computer Systems 121:59- 73.

[5] Ullah, Shamsher, Jiangbin Zheng, Nizamud Din, Muhammad Tanveer Hussain, Farhan Ullah, and Mahwish Yousaf (2023) Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey. Computer Science Review. no.47.

[6] Hesham F.AbdelRazik, Ahmed S.ELSayed, Sally S.Ismail, Abeer M.Mahmoud (2023) Traditional and Deep Learning Techniques in Image Steganography: Recent Advances. Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE.

[7] Jeng-Shyang Pan, Xiao-Xue Sun, Shu-Chuan Chu, Ajith Abraham, Bin Yan. (2021) Digital watermarking with improved SMS applied for QR code. Engineering Applications of Artificial Intelligence 97, p.104049.

[8] Wan, Wenbo, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun (2022) A comprehensive survey on robust image watermarking. Neurocomputing.

[9] Abeer M. Mahmoud, Hanen Karamti, Myriam Hadjouni (2020) A hybrid late fusion-genetic algorithm approach for enhancing CBIR performance. Multimedia. Tools Appl 79:27-28. https://doi.org/10.1007/s11042-020-08825-6

[10] Abeer M. Mahmoud, Hanen Karamti, Fadwa M. Alrowais (2020) A Two Consequent Multi-layers Deep Discriminative Approach for Classifying fMRI Images. Int. J. Artificial. Intelligence. Tools 29(6):2030001:1-2030001:21. https://doi.org/10.1142/S021821302030001X

[11] Chahar, V., Laddha, S., Sharma, A. & Dogra, N (2020) Steganography techniques using convolutional neural networks. Rev. Computation. Eng. Stud 7:66–73.

[12] Hussain, I., Zeng, J., Qin, X. & Tan, S (2020) A survey on deep convolutional neural networks for image steganography and steganalysis. Ksii Trans. Internet Inf. Syst 14:1228–1248.

[13] Zhang, R., Dong, S. & Liu, J (2019) Invisible steganography via generative adversarial networks. Multimedia. Tools Appl 78:8559–8575.

[14] Qin, Jiaohua, Jing Wang, Yun Tan, Huajun Huang, Xuyu Xiang, and Zhibin He (2020) Coverless image steganography based on generative adversarial network. Mathematics 8(9).

[15] ALabaichi, Ashwak, Maisa'A. Abid Ali K. Al-Dabbas, and Adnan Salih (2020) Image steganography using least significant bit and secret map techniques. International journal of electrical & computer engineering 10(1):2088-8708. http://doi.org/10.11591/ijece.v10i1.pp935-946

[16] Biswas, Rajib, and Samir Kumar Bandyapadhay (2020) Random selection-based GA optimization in 2D-DCT domain color image steganography. Multimedia Tools and Applications 79(11).

[17] Baluja, Shumeet (2017) Hiding images in plain sight: Deep steganography. Advances in neural information processing systems 30.

[18] Ray, Biswarup, Souradeep Mukhopadhyay, Sabbir Hossain, Sudipta Kr Ghosal, and Ram Sarkar (2021) Image steganography using deep learning based edge detection. Multimedia Tools and Applications 80(24). https://doi.org/10.1007/s11042-021-11177-4

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

[20] Veerashetty, Sachinkumar (2022) Secure communication over wireless sensor network using image steganography with generative adversarial networks. Measurement: Sensors 24:100452.

[21] Baluja, Shumeet (2019) Hiding images within images. IEEE transactions on pattern analysis and machine intelligence 42(7).

[22] Lu, Shao-Ping, Rong Wang, Tao Zhong, and Paul L. Rosin (2021) Large-capacity image steganography based on invertible neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition pp (10816-10825).

[23] Das, Abhishek, Japsimar Singh Wahi, Mansi Anand, and Yugant Rana (2021) Multi-image steganography using deep neural networks. arXiv preprint arXiv, pp (2101.00350).

[24] Guan, Zhenyu, Junpeng Jing, Xin Deng, Mai Xu, Lai Jiang, Zhou Zhang, and Yipeng Li (2022) DeepMIH: Deep invertible network for multiple image hiding. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(1). 10.1109/TPAMI.2022.3141725

[25] Chen, Haoyu, Linqi Song, Zhenxing Qian, Xinpeng Zhang, and Kede Ma (2022) Hiding images in deep probabilistic models. Advances in Neural Information Processing Systems 35:36776-36788.

[26] Li, Fengyong, Yang Sheng, Xinpeng Zhang, and Chuan Qin (2023) iSCMIS: Spatial-Channel Attention Based Deep Invertible Network for Multi-Image Steganography. IEEE Transactions on Multimedia.

[27] Ping, Ping, Bobiao Guo, Olano Teah Bloh, Yingchi Mao, and Feng Xu (2023) Hiding Multiple Images into a Single Image Using Up-sampling. IEEE Transactions on Multimedia. 10.1109/TMM.2023.3322316

[28] Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (2009) ImageNet: A largescale hierarchical image database. IEEE conference on computer vision and pattern recognition pp (248-255).

[29] D. Martin, C. Fowlkes, D. Tal, and J. Malik (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. in ICCV 2:416 – 423.

[30] Zhang, Mi, and Alexander A. Sawchuk (2012) USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. Proceedings of the ACM conference on ubiquitous computing pp (1036-1043).

[31] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014) Microsoft coco: Common objects in context. Computer Vision–ECCV: 13th European Conference 8693:740-755.

[32] Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman (2009) The pascal visual object classes (voc) challenge. International journal of computer vision 88:303-308.

[33] Agustsson, Eirikur, and Radu Timofte (2017) Ntire 2017 challenge on single image super resolution: Dataset and study. Proceedings of the IEEE conference on computer vision and pattern recognition workshops pp (126-135).

[34] Acharya, Tinku, and Ping-Sing Tsai (2007) Computational foundations of image interpolation algorithms. Ubiquity (4).

[35] Sharma, Raju Pratap, and Aruna Malik (2022) A Comparative Review on Image Interpolation-Based Reversible Data Hiding. In International Conference on Computing, Communications, and Cyber-Security pp (687-699).

[36] Rukundo, Olivier (2023) Normalized weighting schemes for image interpolation algorithms. Applied Sciences 13(3). https://doi.org/10.1007/978-981-99-1479-1_51

[37] Fadnavis, Shreyas (2014) Image interpolation techniques in digital image processing: an overview. International Journal of Engineering Research and Applications 4(10).

[38] Laurent Dinh, David Krueger, and Yoshua Bengio (2014) Nice: Nonlinear independent components estimation. arXiv preprint arXiv:1410.8516.

[39] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio (2017) Density estimation using real NVP. In International Conference on Learning Representations pp (24-26).

[40] Diederik P Kingma and Prafulla Dhariwal (2018) Glow: Generative flow with invertible 1x1 convolutions. arXiv preprint arXiv:1807.03039.

[41] Tycho FA van der Ouderaa and Daniel E Worrall (2019) Reversible Gans for memory-efficient image-to-image translation.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp (4720–4728).

[42] Jing, Junpeng, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan (2021) Hinet: Deep image hiding by invertible network. In Proceedings of the IEEE/CVF international conference on computer vision pp (4733-4742).

[43] Jessica Fridrich, Miroslav Goljan, and Rui Du (2001) Detecting lsb steganography in color, and gray-scale images. IEEE Multimedia, pp (22–28). 10.1109/93.959097

[44] Stephane G Mallat (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11:674–693.

[45] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy (2018) Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV).

[46] Wang, Xintao, Liangbin Xie, Chao Dong, and Ying Shan (2021) Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In Proceedings of the IEEE/CVF international conference on computer vision pp (1905-1914).

[47] El-Assiouti, Hosam S., Hadeer El-Saadawy, Maryam N. Al-Berry, and Mohamed F. Tolba (2023) Lite-SRGAN and Lite-UNet: Towards Fast and Accurate Image Super-Resolution, Segmentation, and Localization for Plant Leaf Diseases. IEEE Access. 10.1109/ACCESS.2023.3289750

[48] Zhao, Yafeng, Zhen Chen, Xuan Gao, Wenlong Song, Qiang Xiong, Junfeng Hu, and Zhichao Zhang (2021) Plant disease detection using generated leaves based on DoubleGAN. IEEE/ACM transactions on computational biology and bioinformatics 19(3).

[49] Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition pp (4681-4690).

[50] Justin Johnson, Alexandre Alahi, and Li Fei-Fei (2016) Perceptual losses for real-time style transfer and super-resolution. In ECCV. https://doi.org/10.1007/978-3-319-46475-6_43

[51] Diederik, P. Kingma (2014) Adam: A method for stochastic optimization.

[52] Elharrouss, Omar, Noor Almaadeed, Somaya Al-Maadeed (2020) An image steganography approach based on k-least significant bits (k-LSB). IEEE international conference on informatics, IoT, and enabling technologies (ICIoT). 10.1109/ICIoT48696.2020.9089566

[53] X. Weng, Y. Li, L. Chi, and Y. Mu (2019) High-capacity convolutional video steganography with temporal residual modeling. In Proc. Int. Conf. Multimedia Retrieval pp (87–95).

[54] Sara, U., Akter, M. & Uddin, M (2019) Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study. J. Computation. Communication p.73002.

[55] Hore, A. & Ziou, D (2010) Image quality metrics: PSNR vs. SSIM. In Proceedings of IEEE International Conference on Pattern Recognition pp (2366–2369).

[56] Tsai, D. Y., Lee, Y. & Matsuyama, E (2008) Information entropy measure for evaluation of image quality. J. Digit. Imaging 21:pp338–347. https://doi.org/10.1007/s10278-007-9044-5