Received June 29, 2021, accepted September 10, 2021, date of publication September 20, 2021, date of current version October 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113953

End-to-End Image Steganography Using Deep Convolutional Autoencoders

NANDHINI SUBRAMANIAN[1], (Member, IEEE), ISMAHANE CHEHEB[2], (Member, IEEE), OMAR ELHARROUSS[1], (Member, IEEE), SOMAYA AL-MAADEED[1], (Senior Member, IEEE), AND AHMED BOURIDANE[3], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[2]Department of Computer and Information Science, Northumbria University, Newcastle upon Tyne NE1 8QH, U.K.
[3]Centre for Data Analytics and Cybersecurity, University of Sharjah, Sharjah, United Arab Emirates

Corresponding author: Nandhini Subramanian (ns1808900@qu.edu.qa)

Open Access funding was provided by the Qatar National Library.

ABSTRACT Image steganography is used to hide a secret image inside a cover image in plain sight. Traditionally, the secret data is converted into binary bits and the cover image is manipulated statistically to embed the secret binary bits. Overloading the cover image may lead to distortions and the secret information may become visible. Hence the hiding capacity of the traditional methods are limited. In this paper, a light-weight yet simple deep convolutional autoencoder architecture is proposed to embed a secret image inside a cover image as well as to extract the embedded secret image from the stego image. The proposed method is evaluated using three datasets - COCO, CelebA and ImageNet. Peak Signal-to-Noise Ratio, hiding capacity and imperceptibility results on the test set are used to measure the performance. The proposed method has been evaluated using various images including Lena, airplane, baboon and peppers and compared against other traditional image steganography methods. The experimental results have demonstrated that the proposed method has higher hiding capacity, security and robustness, and imperceptibility performances than other deep learning image steganography methods.

INDEX TERMS Image steganography, deep learning, autoencoder, information hiding.

I. INTRODUCTION

Technology is steering the social and economic developments. With the introduction of the Internet of Things (IoT), the necessity for the transfer of data between the sensors, cloud servers, and end devices has become inevitable. There is a complete changeover in the modes and forms of communication and digital media has become the primary mode of communications. Even in offices, people have shifted from using paper-based memos to electronic mails for communication. With the unlimited computational power and storage, transfer of images and videos have become possible. Images and videos are preferred as the predominant form of communication. On the other side, with all these technological developments, the need for protecting the privacy, security, and confidentiality of the data that is being transferred arises. Information hiding techniques are potentially a useful rescue to facilitate the secure transfer of data.

Information security methods can be classified into three categories: digital watermarking, steganography, and cryptography. Cryptography is the most widely used method and works by converting the plaintext into ciphertext. The ciphertext generated by a cryptographic protocol is visible to human eyes leading to suspicion. Image steganography is the art and science of hiding secret information inside an image file. It is a research field that is gaining popularity to secure data transfer or storage. On the other hand, image steganalysis can be used to break and uncover the embedded secret information from the image. Steganography methods can deal with a variety of digital media like images, videos, text and audio. The data structure of the secret message is not modified and the hidden message is not visible, hence reducing any suspicion and scrutiny.

Image steganography is correlated to the prisoners' problem, where Alice and Bob are in prison. They want to escape, but, any communication between them is monitored by Eve. Eve is suspicious of Alice and Bob and so scrutinizes all the communication between them. In this case, Alice and Bob

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao.

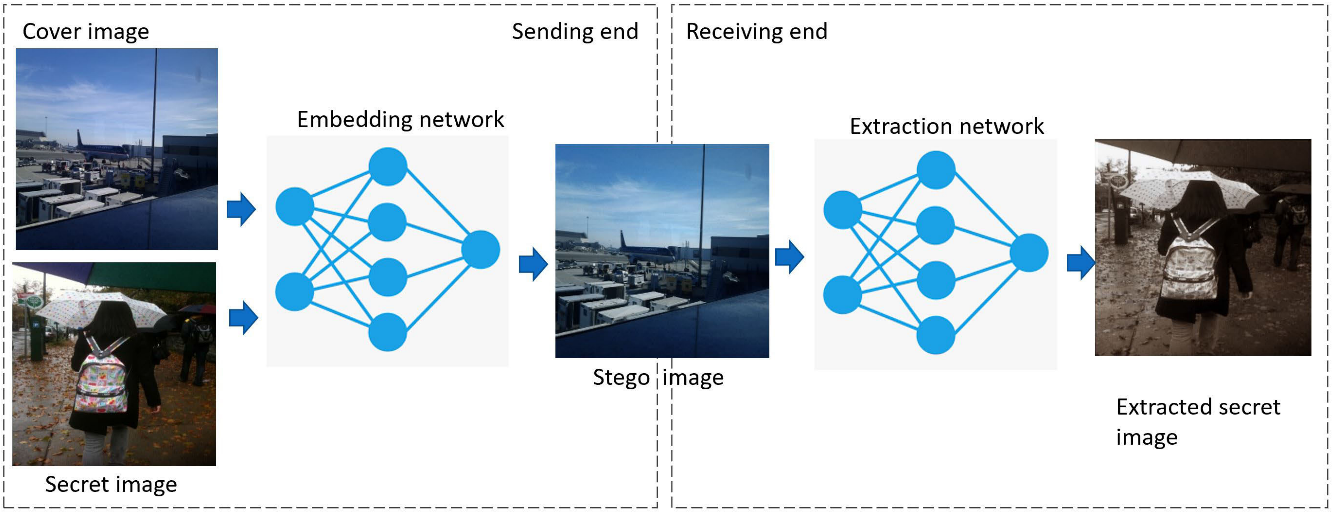

FIGURE 1. Overall workflow of the proposed method. The sending end consists of the preprocessing module and the embedding network. The preprocessing module extracts features from the cover image and the secret image and merges them using the concatenation layer. The embedding network reconstructs the stego image from the fused features. Finally, at the receiving end is the extraction network to extract the secret image from the stego image.

should communicate in a way that is understandable to only them. For Eve it should look like normal message [1].

Image steganography has attracted significant interests from the research community resulting in considerable enhancements in line with the recent advances of digital media technologies. Initially, traditional methods such as Least Significant Bits (LSB) substitution, Pixel Value Differencing (PVD), Discrete Wavelet Transformation (DWT) were used for hiding the secret information inside a carrier image by exploiting the pixel value of the cover image [2]. However, the hiding capacity of the traditional methods is limited due to embedding capacity issues which can lead to the exposure of the presence of the secret media. Deep learning is another paradigm which is extensively used in computer vision applications. Deep learning methods including autoencoders and generative adversarial networks (GANs) have been adapted and employed in image steganography. These methods have increased the hiding capacity, security and robustness compared to traditional methods. Reversible image steganography is a technique where steganography and steganalysis are performed together [3]. Reversible image steganography using deep learning method has further increased hiding capacity, security and robustness. In this paper, an autoencoder-decoder architecture model is proposed for performing steganography and steganalysis. The main aim of this paper is to design a reversible end-to-end image steganography model. The contribution of this work is listed as follows:

- A simple and light weight autoencoder-decoder model architecture is used for embedding and extracting the secret image inside the cover image.
- Instead of concatenating the raw images, features extracted from the cover image and secret image are concatenated. This reduces the amount of redundant data in the raw images and uses only the most important features of the images.
- A customized loss function is designed and exchanged as feedback to the training model to optimize the learning function.

II. RELATED WORKS

Steganography is not a new topic and has been in existence since 440 BC. It has historically evolved from shaving the heads of the sub-ordinates, engraving the secret message using invisible inks during the world war to digital steganography in recent times. Statistical methods including the LSB substitution, PVD methods are used to hide secret messages in cover images. In the LSB method, the secret information is converted into binary bits which are embedded in the least significant bits of the carrier image [4]. It is assumed that the resolution of the cover image is high and exploiting the three least significant bits for every byte in the image will not reduce the precision or arouse any suspicion [5] and [6]. Another variant of LSB uses Huffman encoding on the secret information before embedding [7]. PVD is another statistical method used to hide larger number of binary bits of the secret information in the edges of a cover image and smaller number in the smoother regions of a cover image [8].

Information hiding on quantum images with modification of the direction technique [9], [10], local binary patterns [11] are some of the other techniques used. Most of the traditional methods can handle only text as the secret message with reduced hiding capacity. Reverse engineering is possible to attack the steganography as the embedding happens by statistically exploiting the cover image. A combination of cryptography and steganography is also proposed to increase the overall anti-detection property [10]. Integer wavelet transform (IWT) is applied on the cover image and the chaotic map is used to determine the pixels for embedding the binary bits of the secret message [12]. An inverse integer wavelet transform is performed to produce the stego image. During the extraction, the chaotic map used for embedding is generated again and the secret message is extracted with the help of the generated chaotic map. Similarly, a 3D sine chaotic map is used by Valandar *et al.* [13]. Framelet transformation is applied on the cover image to recognize the transformer coefficients to embed the secret message bits without causing any visual distortion. A novel method called the Pixel Density Histogram (PHD) is proposed for halftone images [14], [15].

Deep learning (DL) methods, which have been widely used recently, have emerged as a viable tool in image steganography applications showing attractive performance and efficiency. Typically, CNN-based architectures and GAN-based methods are used for performing end-to-end steganography and steganalysis. The Autoencoder model is a popular network architecture used in image steganography [3], [16]–[18], and [19]. Three components are needed for an end-to-end image steganography and steganalysis [20], [21]: (i) the preparation of the input images - cover and secret images, (ii) the hiding network for embedding the secret image inside the cover image and (iii) the reveal network for extracting the secret image from the container steganography image. Pixelwise CNN [22] and stylenet [23] are other networks used for the implementation of image steganography. To improve the anti-detection property of the steganography algorithm, the secret image is first transformed using DCT and then encrypted using Elliptic Curve Cryptography(ECC) [24]. A SegNet architecture is used as the backbone for hiding and extraction networks. The input to the hiding network is the encrypted secret image and the cover image and the output form the container image. The container is the input image for the extraction network and the output is the encrypted version of the secret image. Finally, the decryption algorithm is carried out on the revealed image to obtain the final secret image. A Pyramid Pooling layer is placed in between the down sampling and the up sampling block and an ablation study is conducted to prove that the addition of pyramid pooling layer increases the performance of the model [25]. Generative Adversarial Networks (GANs) are also widely used in the field of image steganography [26] and various GAN architectures have been proposed; for example, DCGAN [27], [28], WGAN [29], [30], cycleGAN [31]–[33]. Embedding simulators in the place of steganalyzers are also implemented [34], [35]. A sender-receiver scheme [36], [37], coverless steganography [3], [38], [39], and cryptography-scheme based [40], [41] are some of the other variations on the implementation of image steganography with GAN model as the base. A detailed description of the recent advances in image steganography can be found in [2].

The major shortcoming in the traditional image steganography method is the low hiding capacity. Trying to hide more information by tweaking a greater number of bits in the cover may expose the hidden secret information. Since the hiding happens by statistically exploiting the pixel values, steganalysis can be easily performed by reverse engineering. This will affect the security of the method. The quality of the extracted secret information may also be subsided thus affecting the robustness. Yet another drawback is that the media of the secret communication is mostly text. Even if an image is used, only grayscale images are used. Hiding the pixel values of the three-channel secret image inside another three-channel cover image can get quite difficult in traditional methods. However, the hiding capacity, which is an issue with most of the traditional methods, can be solved using deep learning based methods. The capacity is increased at the cost of the storage space, memory, and computation time because of the complexity of the models. In this paper, a simple autoencoder architecture that can hide a secret image inside a cover image is proposed with increased hiding capacity. In addition to an increased hiding capacity, security and robustness of the proposed method are also higher than the traditional methods.

III. PROPOSED METHODOLOGY

The overall workflow of the proposed method is given in figure 1 and consists of three modules - preprocessing module, embedding network and the extraction network. The preprocessing module prepares the cover image and secret image for the embedding network to reconstruct the stego image. The purpose of the embedding network is to reconstruct the stego image which hides the secret image inside the cover one. The extraction network recovers the hidden secret image from the container stego image. The preprocessing module together with the embedding network is placed at the sending end to produce the stego image. The extraction network is deployed at the receiving end to extract the secret image from the stego image. More details on each of the modules are given in the below subsections.

Mathematically, the proposed solution can be expressed as follows. Let c be the cover image and s be the secret image, the preprocessing module produces features f(c) and f(s) for the cover and the secret image. The final output of the preprocessing module is the aggregate of the features extracted $f(c)+f(s)$. The main aim of the embedding method is to produce a stego image c' such that $c' \approx$ c and extraction network is to extract the secret image s' which is $s' \approx$ s.

A. PREPROCESSING MODULE

Instead of processing the raw form of the cover and the secret images, features are extracted from them using the preprocessing module. High resolution images often contain redundant data and by extracting the most meaningful features, the burden on the embedding network is reduced. The input size should be of the format m × m × n, which represents the three dimensions - width, height and depth. The width and height should be of the same size hence they are represented by m. After a thorough analysis of the existing literature [3], [21], [24], [39], the input size of the cover image is fixed to be 256 × 256. The input secret image can be of any size, the preprocessing module resizes the secret image

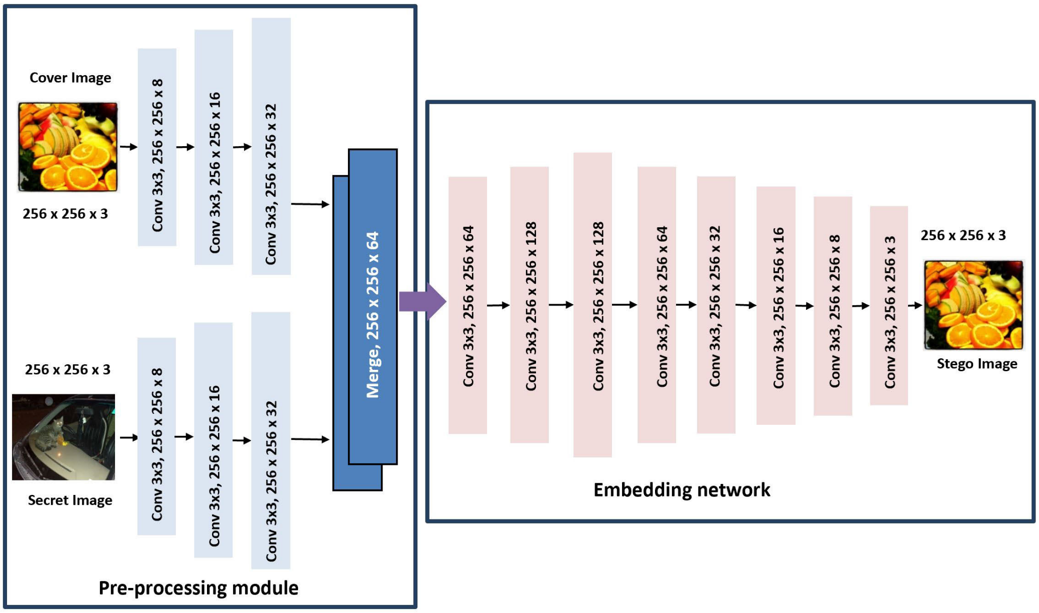

FIGURE 2. The architecture of the preprocessing module and the embedding network of the proposed method.

to 256 × 256 since the cover image and the secret image should be of same size. The resize function from the skimage library is used to resize the cover image and the secret image to a fixed size of 256 × 256. Instead of representing the input images as colour gradients, the preprocessing module converts them into useful features that can be used by the embedding network. The preprocessing module consists of one input layer and three convolutional layers with increasing number of filters. The choice of the number of filters, filter size and the stride are purely dependent on the application. The main purpose of the preprocessing module is to extract usable and meaningful features through convolutional layers with different filter sizes. Initially, lower-level local features such as edges are extracted by using smaller filter sizes. The filter size is increased to help the model learn more sophisticated features. The number of filters used are 8, 16 and 32. The cover image and the secret image are passed through the preprocessing module in parallel. Finally, a merge layer is designed which concatenates the features extracted from the cover image and the secret image.

B. EMBEDDING NETWORK

The preprocessing module and the embedding network together are designed based on an auto-encoder architecture concept. The embedding network along with the preprocessing module have a hourglass structure with an expanding phase and a contracting phase. The autoencoder network takes the input and extracts the features using the encoder part. The latent space in an autoencoder is the feature representation of the input. The decoder part of the autoencoder is used to reconstruct the output image from the latent space. Image steganography applications does not require any dimensionality changes, the latent space should be the combined feature representation of the cover image and the secret image. The embedding network takes the concatenated features from the preprocessing module as the input to produce a latent space and reconstruct the stego image (which is close in resemblance to the cover image) from the latent space. Every bit of the secret image is hidden across every available bit of the cover image. The embedding network is designed with two convolutional layers with an increasing number of filters. The latent space at the end of the encoder represents the finer features of both cover image and the secret image concatenated. The decoder part of the embedding network has five convolutional layers with a decreasing number of filters since there is no need for any dimensionality change(s). The number of filters in the encoder part are 64, 128 and the decoder part of the embedding network has 128, 64, 32, 16 and 8 filters. ReLU activation is added at the end of the convolutional layers to introduce linearity by giving the max value for positives and 0s for negatives. ReLU is used because it makes the training easier with better performance as it overcomes the vanishing gradient problem which is common in architectures with multiple layers. ReLU can be given as h(c) = max(0,c). A convolutional layer with 3 filters is placed at the end of the embedding network to convert the 256 × 256 × 8 feature vector into 256 × 256 × 3 stego image output. Figure 2 represents the architecture of the preprocessing module and the embedding network together.

C. EXTRACTION NETWORK

The extraction network aims to extract the secret image hidden inside the stego image. After conducting controlled experiments, an architecture identical to the embedding network seems to give the best results in extracting the secret image with minimum information loss. The extraction network has an expanding phase and a contracting phase. The number of filters, filter size, stride and other hyper-parameters are fine-tuned based on the experimental results. The architecture which produced the best result is described here. The expanding encoder part of the extraction network has five convolutional layers with an increasing number of filters (8, 16, 32, 64, 128). The decoder part has five convolutional layers with an decreasing number of filters (128, 64, 32, 16, 8). Each layer is designed with an ReLU activation. The decoder of the extraction network is followed by a convolutional layer with 3 filters to construct the extracted secret image. The extraction network architecture is given in figure 3.

D. CUSTOMIZED LOSS FUNCTION

Unlike conventional image reconstruction, image steganography process requires two input images and two output images. Therefore regular loss function may not be suitable for this purpose. A customized loss function is introduced to increase the performance of the architecture. There are two losses to be calculated: the embedding loss and the extraction loss. The embedding loss is calculated between the input cover image and the output stego image produced by the embedding network. On the other hand, the extraction loss is calculated between the input secret image and the extracted secret image by the extraction network. The overall loss is the sum of the embedding and extraction loss.

Let i be the cover image and i' the reconstructed cover image with the secret image generated by the embedding network. Also, let h be the secret image and h' the extracted secret image by the extraction network. The loss function has to be customized in such a way that it will help the model to optimize the learning function. Loss is a feedback measure given back to the model while training in each epoch as a measure of how well the model is performing through back-propagation.

The loss of the embedding network, L_{emb}, is given by equation 1 and the loss of the extraction network, L_{ext}, is given by equation 2. Finally, the overall loss, L, is calculated using equation 3.

$$L_{emb} = |i - i^{|} \tag{1}$$

$$L_{ext} = |h{-}h'| \tag{2}$$

$$L = L_{emb} + \alpha * L_{ext} = |i - i'| + \alpha * |h - h'| \tag{3}$$

where α is the error adjustment and is fixed to 0.3.

Initial experiments were conducted by varying the values of α from 0.3, 0.6 and 0.9. Increasing the value of α increased the loss and 0.3 value produced optimal loss value. The embedding network's loss function is given back to the embedding network and the overall loss is given to the extraction network to minimize the distortions of the extracted secret image.

TABLE 1. Details on the datasets.

Dataset	Total images	Purpose
ImageNet	15 M	Object detection and localization, image classification
COCO	328 K	Image classification, Object recognition and segmentation
CelebA	200 M	Face recognition, face detection

IV. EXPERIMENTAL SETUP

The experiments were conducted on ASUS laptop with Intel CORE i7, NVIDIA GEFORCE graphical card. Python 3 with Keras library using Tensorflow backend is the programming language employed throughout the experiments. Adam optimizer is used and the model was trained for 5 epochs.

Three important factors have to be evaluated for analysing the performance of the steganography model: hiding capacity, security and robustness and the imperceptibility. The hiding capacity is defined as the amount of secret information that can be hidden without distorting the cover image. In other words, this is the capacity per pixel of the steganography model. Since the cover image and the secret image have the same size (256 × 256), the capacity of the proposed method is 1. The hiding capacity can be calculated using equation 4.

$$capacity = \frac{L}{H * W * C} \tag{4}$$

where L is the length of the secret information. In this case, it is the product value of the height, width and channel of the secret image. H,C and W represent the height, number of channels and width of the cover image.

Security is the ability to hide the data which can only be accessed by authorized users while the robustness is the ability of the model to embed and retrieve the secret media without distortions. Peak Signal-to-Noise Ratio (PSNR) is a popular metric used to measure the similarities between two images. First, Mean Squared Error (MSE) is calculated and the PSNR value is calculated from MSE. Equations to calculate the MSE and PSNR are given in equation 5 and 6 respectively.

$$MSE = \frac{\sum_{R,C}[I_1(r,c) - I_2(r,c)]}{R * C} \tag{5}$$

$$PSNR = 10 * \log_{10} \frac{E_2}{MSE} \tag{6}$$

Imperceptibility is a measure used to verify the visibility of the secret message hidden. Image results produced by the proposed model against the test set of each dataset is given to evaluate the imperceptibility.

Three datasets are used for training and testing the performances of the proposed architecture. Table 1 provides the necessary details on the datasets used. 45000 image pairs from the datasets are taken for training and 5000 image pairs are used for testing. From the training images pair, 80 % is chosen for training and 20% is chosen for validation in random.

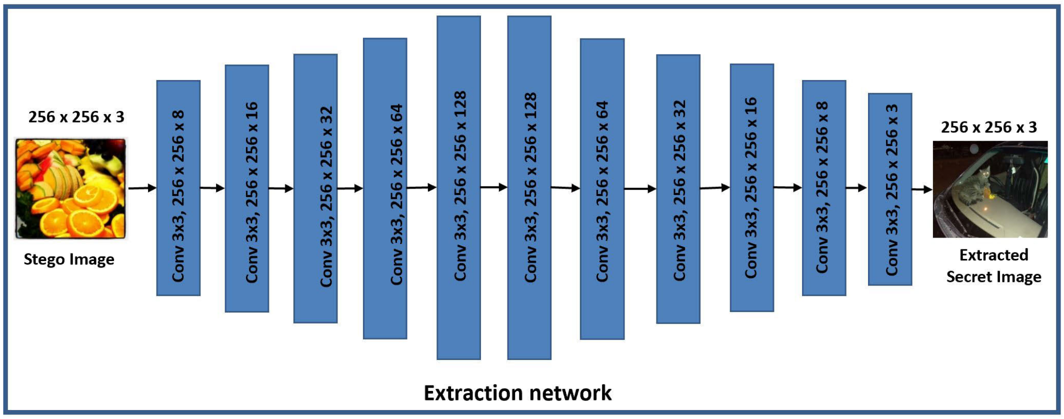

FIGURE 3. The architecture of the extraction network of the proposed method.

V. RESULTS AND DISCUSSION

The capacity of the proposed model is calculated and is then compared with a few other similar image steganography methods. It can be seen from the results that the hiding capacity of the proposed architecture is 1, which means, every bits of the secret image is hidden inside every bit of the cover image.

Some qualitative images on the test sets of the three datasets - COCO, ImageNet, CelebA are shown in figure 4. The images clearly show a higher imperceptibility of the proposed method. Even with increased hiding capacity, the secret image is hidden without any distortions to the cover image. The secret image is extracted back as well without much information loss.

The computational complexity is another important factor to be evaluated. Sometimes, based on the application scenario, a compromise between efficiency and computational complexity is made. A critical comparison of the time taken for training the embedding network on all the three datasets along with the computation time taken for the trained model to load and generate a single stego image is made in the bar chart shown in figure 5. A similar comparison for the extraction network is made in figure 5. The time taken by the embedding network is approximately twice the time taken by the extraction network. The reason is that the embedding network process two images (cover and secret), whereas, the extraction network processes only one image (stego image).

MSE and the PSNR values of the proposed method are reported and compared against state-of-the-art methods in table 4 and 2 respectively. The proposed method has comparatively better PSNR values compared with other deep learning methods [18], [21], [42]. Another important aspect to be noted is that the proposed method is the lightest architecture in comparison. As shown in the table, the methods in comparison [18], [21], [42] use grayscale images as the secret media. Thus the hiding capacity is less. The proposed method uses RGB image as the secret media and so the hiding capacity is increased three-folds.

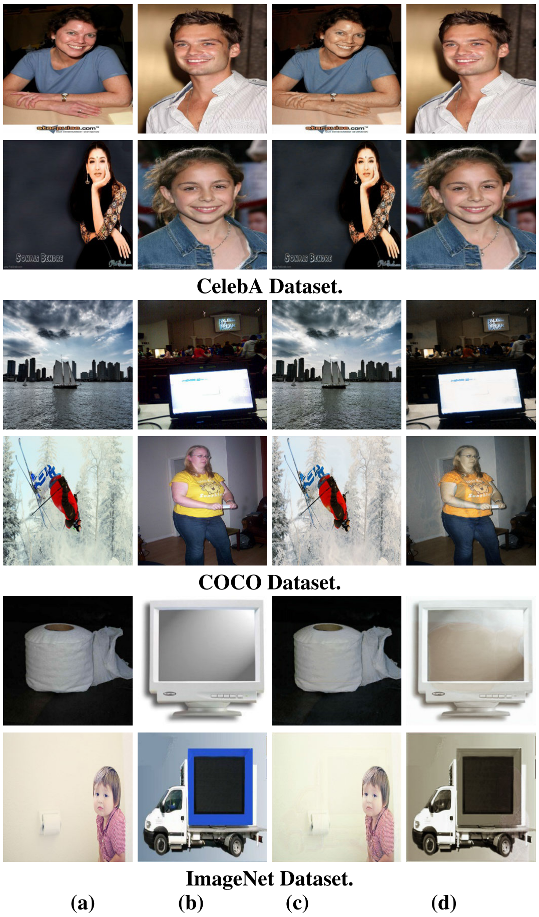

FIGURE 4. Qualitative image results of the proposed model on the CelebA, COCO and ImageNet datasets. (a) represents the cover image, (b) represents the secret image, (c) represents the stego image of (b) embedded in (a), and (d) represents the extracted secret image from (c).

In order to compare the proposed method against traditional methods, some popular images from SIPI dataset such as lena, airplane, fishing boat, truck, house, baboon and peppers are used. The same images are passed through

TABLE 2. Result comparison between the proposed method and other deep learning methods.

Method	Technique	Cover Image Size	Secret Image Size	Payload(%)	PSNR
Rehman's method [18]	Encoder-decoder	32 × 32 × 3	32 × 32 × 1	33	29.6
Zhang's method [21]	GAN	256 × 256 × 3	256 × 256 × 1	33	33.92
Zhang's method (ISGAN) [21]	GAN	256 × 256 × 3	256 × 256 × 1	33	34.01
Chen's method [42]	ISGAN	300 × 300 × 3	300 × 300 × 1	33	34.07
Proposed Method	Encoder-decoder	256 × 256 × 3	256 × 256 × 3	100	**34.55**

TABLE 3. Result comparison between the proposed method and other traditional methods.

Method	Cover image size	Secret image size	Technique	Capacity (bpp)	PSNR (dB)
[43]	512 × 512 × 3	256 × 256	DCT	20.83	27.24
[44]	512 × 512 × 1	Text	DCT	1.02	28.00
[45]	512 × 512 × 1	Text	DCT	1.96	28.00
[46]	512 × 512 × 3	470 × 470	DCT	22.52	28.16
[46]	512 × 512 × 3	470 × 470	DCT	21.71	29.20
[47]	512 × 512 × 3	256 × 256 × 3	LSB	-	31.99
[10]	513 × 513 × 3	513 × 513 × 3	LSB and MSB modification	-	32.09
[4]	512 × 512 × 3	512 × 512 × 3	k-LSB	-	32.44
[48]	512 × 512 × 3	Text	PVD,QVD and LSB	4.55	33.06
[49]	129 × 129 × 3	Text	LSB on edges	2.9749	34.33
Proposed Method	256 × 256 × 3	256 × 256 × 3	DL	24	33.70

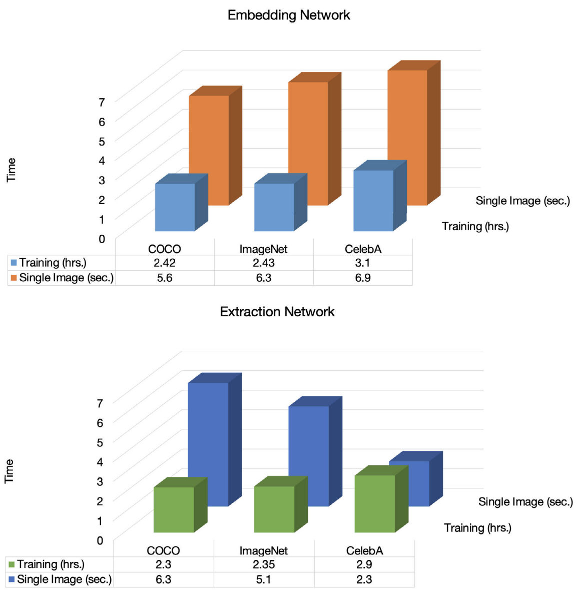

FIGURE 5. Critical time analysis on (a) Embedding network, and (b) Extraction network.

TABLE 4. MSE and PSNR values of the proposed method.

Dataset	Image size	bpp	Network	MSE	PSNR
COCO	256 × 256	24	Embedding	44.01	31.96
			Extraction	105.37	27.90
ImageNet	256 × 256	24	Embedding	51.97	34.55
			Extraction	104.92	27.93
CelebA	256 × 256	24	Embedding	41.15	32.26
			Extraction	105.10	27.92

the proposed model and PSNR value of the stego image generated for each image is calculated. The average value of the PSNR is calculated and used for comparison against other traditional methods. The capacity of the traditional methods and the proposed method is calculated using equation 4. For traditional methods, the value of L in equation 4 is the amount of secret bits in case of text. When the secret media is an image, the value of L is the product of length, width and height of the image. The average PSNR results obtained from the proposed methods on these common images are given and compared with other traditional methods in table 3. The proposed method has only marginally high PSNR values with the methods in comparison. However, the hiding capacity is the highest for the proposed method and the secret media used is the RGB image.

From the results shown, it can be clearly seen that the proposed architecture has higher security, robustness, imperceptibility and information hiding capacity.

VI. CONCLUSION

In this paper, a light-weight but simple architecture is proposed to achieve end-to-end image steganography. The proposed architecture is inspired from the deep convolutional variation of the autoencoder. The whole system comprises of preprocessing module, embedding network and the extraction network. The preprocessing module prepares the input images for the embedding network to hide the secret image inside the cover image. The extraction network extracts the secret image from the stego image produced by the embedding network. The PSNR value of the proposed method is higher showing the higher security and robustness of the proposed method compared to other traditional and deep learning image steganography methods. The capacity of the proposed method is 1 and is the highest when compared with the traditional methods. The proposed method has an upper hand in terms of invisibility as well and can produce stego images very similar to the input cover image.

ACKNOWLEDGMENT

This work was made possible by NPRP11S-0113-180276 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the author. Open Access funding was provided by the Qatar National Library.

REFERENCES

[1] R. Böhme, *Principles of Modern Steganography and Steganalysis*. Berlin, Germany: Springer, 2010, pp. 11–77.

[2] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image steganography: A review of the recent advances," *IEEE Access*, vol. 9, pp. 23409–23423, 2021.

[3] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a U-Net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.

[4] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)," in *Proc. IEEE Int. Conf. Informat., IoT, Enabling Technol. (ICIoT)*, Feb. 2020, pp. 131–135.

[5] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, Feb. 1998.

[6] S. Gupta, G. Gujral, and N. Aggarwal, "Enhanced least significant bit algorithm for image steganography," *Int. J. Comput. Eng. Manage.*, vol. 15, no. 4, pp. 40–42, 2012.

[7] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman encoding," in *Proc. 3rd Nat. Conf. Emerg. Trends Appl. Comput. Sci.*, Mar. 2012, pp. 14–18.

[8] H.-S. Huang, "A combined image steganographic method using multi-way pixel-value differencing," in *Proc. 6th Int. Conf. Graphic Image Process. (ICGIP)*, Mar. 2015, pp. 267–271.

[9] S. Wang, J. Sang, X. Song, and X. Niu, "Least significant qubit (LSQb) information hiding algorithm for quantum image," *Measurement*, vol. 73, pp. 352–359, Sep. 2015.

[10] N. Patel and S. Meena, "LSB based image steganography using dynamic key cryptography," in *Proc. Int. Conf. Emerg. Trends Commun. Technol. (ETCT)*, Nov. 2016, pp. 1–5.

[11] A. Qiu, X. Chen, X. Sun, S. Wang, and W. Guo, "Coverless image steganography method based on feature selection," *J. Inf. Hiding Privacy Protection*, vol. 1, no. 2, p. 49, 2019.

[12] M. Y. Valandar, P. Ayubi, and M. J. Barani, "A new transform domain steganography based on modified logistic chaotic map for color images," *J. Inf. Secur. Appl.*, vol. 34, pp. 142–151, Jun. 2017.

[13] M. Y. Valandar, M. J. Barani, P. Ayubi, and M. Aghazadeh, "An integer wavelet transform image steganography method based on 3D sine chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 8, pp. 9971–9989, Apr. 2019.

[14] W. Lu, Y. Xue, Y. Yeung, H. Liu, J. Huang, and Y. Shi, "Secure halftone image steganography based on pixel density transition," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1137–1149, May/Jun. 2019.

[15] C. Kim, D. Shin, L. Leng, and C.-N. Yang, "Separable reversible data hiding in encrypted halftone image," *Displays*, vol. 55, pp. 71–79, Dec. 2018.

[16] P. Wu, Y. Yang, and X. Li, "Image-into-image steganography using deep convolutional network," in *Proc. 19th Pacific-Rim Conf. Multimedia Hefei China Adv. Multimedia Inf. Process. (PCM)*, Sep. 2018, pp. 792–802.

[17] P. Wu, Y. Yang, and X. Li, "StegNet: Mega image steganography capacity with deep convolutional network," *Future Internet*, vol. 10, no. 6, p. 54, Jun. 2018.

[18] R. Rahim and S. Nadeem, "End-to-end trained cnn encoder-decoder networks for image steganography," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–6.

[19] N. Subramanian, O. E. Harrouss, and S. El-Seoud, "Image steganography using auto encoder-decoder based deep learning method," in *Proc. Int. Conf. Interact. Collaborative Blended Learn.*, Feb. 2021, pp. 520–530.

[20] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2069–2079.

[21] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8559–8575, Apr. 2019.

[22] K. Yang, K. Chen, W. Zhang, and N. Yu, "Provablysecure generative steganography based on autoregressive model," in *Proc. Int. Workshop Digit. Watermarking*. South Korea: Springer, 2018, pp. 55–68.

[23] Z. Wang, N. Gao, X. Wang, J. Xiang, and G. Liu, "STNet: A style transformation network for deep image steganography," in *Proc. Int. Conf. Neural Inf. Process.* China: Springer, 2019, pp. 3–14.

[24] X. Duan, D. Guo, N. Liu, B. Li, M. Gou, and C. Qin, "A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network," *IEEE Access*, vol. 8, pp. 25777–25788, 2020.

[25] X. Duan, W. Wang, N. Liu, D. Yue, Z. Xie, and C. Qin, "StegoPNet: Image steganography with generalization ability based on pyramid pooling module," *IEEE Access*, vol. 8, pp. 195253–195262, 2020.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[27] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, Jan. 2020, Art. no. 114333.

[28] D. Volkhonskiy, B. Borisenko, and E. Burnaev, "Generative adversarial networks for image steganography," in *Proc. ICRL Conf.*, France, 2016.

[29] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "SSGAN: Secure steganography based on generative adversarial networks," in *Proc. Pacific Rim Conf. Multimedia*. China: Springer, 2017, pp. 534–544.

[30] H. Shi, X.-Y. Zhang, S. Wang, G. Fu, and J. Tang, "Synchronized detection and recovery of steganographic messages with adversarial learning," in *Proc. Int. Conf. Comput. Sci.* Portugal: Springer, 2019, pp. 31–43.

[31] P. G. Kuppusamy, K. C. Ramya, S. Sheebha Rani, M. Sivaram, and V. Dhasarathan, "A novel approach based on modified cycle generative adversarial networks for image steganography," *Scalable Comput., Pract. Exper.*, vol. 21, no. 1, pp. 63–72, Mar. 2020.

[32] R. Meng, Z. Zhou, Q. Cui, X. Sun, and C. Yuan, "A novel steganography scheme combining coverless information hiding and steganography," *J. Inf. Hiding Privacy Protection*, vol. 1, no. 1, p. 43, 2019.

[33] C. Chu, A. Zhmoginov, and M. Sandler, "CycleGAN, a master of steganography," 2017, *arXiv:1712.02950*. [Online]. Available: http://arxiv.org/abs/1712.02950

[34] J. Yang, D. Ruan, J. Huang, X. Kang, and Y.-Q. Shi, "An embedding cost learning framework using gan," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 839–851, 2020.

[35] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Process. Lett.*, vol. 24, no. 10, pp. 1547–1551, Oct. 2017.

[36] X. Zhao, C. Yang, and F. Liu, "On the sharing-based model of steganography," in *Proc. Int. Workshop Digit. Watermarking*, Feb. 2021, pp. 94–105.

[37] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 657–672.

[38] M.-m. Liu, M.-q. Zhang, J. Liu, Y.-n. Zhang, and Y. Ke, "Coverless information hiding based on generative adversarial networks," 2017, *arXiv:1712.06951*. [Online]. Available: http://arxiv.org/abs/1712.06951

[39] X. Duan, H. Song, C. Qin, and M. K. Khan, "Coverless steganography for digital images based on a generative model," *Comput., Mater. Continua*, vol. 55, no. 3, pp. 483–493, Jul. 2018.

[40] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "SSteGAN: Self-learning steganography based on generative adversarial networks," in *Proc. Int. Conf. Neural Inf. Process.* Cambodia: Springer, 2018, pp. 253–264.

[41] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1954–1963.

[42] K. Alex Zhang, A. Cuesta-Infante, L. Xu, and K. Veeramachaneni, "SteganoGAN: High capacity image steganography with GANs," 2019, *arXiv:1901.03892*. [Online]. Available: http://arxiv.org/abs/1901.03892

[43] T. Rabie and I. Kamel, "High-capacity steganography: A global-adaptive-region discrete cosine transform approach," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6473–6493, 2017.

[44] C.-C. Lin and P.-F. Shiu, "High capacity data hiding scheme for dct-based images," *J. Inf. Hiding Multimedia Signal Process.*, vol. 1, no. 3, pp. 220–240, 2010.

[45] B. Yang, M. Schmucker, W. Funk, C. Busch, and S. Sun, "Integer DCT-based reversible watermarking for images using companding technique," *Proc. SPIE*, vol. 5306, pp. 405–415, Jun. 2004.

[46] T. Rabie, I. Kamel, and M. Baziyad, "Maximizing embedding capacity and stego quality: Curve-fitting in the transform domain," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8295–8326, 2018.

[47] R. Nur, "An approach of securing data using combined cryptography and steganography," *Int. J. Math. Sci. Comput.*, vol. 6, no. 1, pp. 1–9, Feb. 2020.

[48] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2995–3004, Apr. 2019.

[49] S. K. Ghosal, A. Chatterjee, and R. Sarkar, "Image steganography based on kirsch edge detection," *Multimedia Syst.*, vol. 27, pp. 73–78, Feb. 2020.

NANDHINI SUBRAMANIAN (Member, IEEE) received the bachelor's degree in electrical and electronics engineering from the PSG College of Technology, India, and the master's degree in computing from Qatar University, Doha. She is currently working as a Research Assistant with Dr. Somaya Al-Maadeed at Qatar University. Her interests include computer vision, artificial intelligence, machine learning, and cloud computing. She won the first rank (Track-2) in the National-Level Artificial Intelligence Competition (Qatar).

ISMAHANE CHEHEB (Member, IEEE) received the master's degree in mathematics and computer science from the University of Sciences and Technology Houari Boumediene (USTHB), Algiers, Algeria, and the Ph.D. degree in computer science from the University of Northumbria, Newcastle upon Tyne, U.K. She is currently a Postdoctoral Fellow with the University of Northumbria. Her research interests include biometrics, image and video processing, and multimedia security.

OMAR ELHARROUSS (Member, IEEE) received the master's degree from the Faculty of Sciences, Dhar El Mehraz, Fez, Morocco, in 2013, and the Ph.D. degree from the LIIAN Laboratory, USMBA-Fez University, in 2017. His research interests include pattern recognition, image processing, and computer vision.

SOMAYA AL-MAADEED (Senior Member, IEEE) received the Ph.D. degree in computer science from Nottingham, U.K., in 2004. She is currently the Coordinator of the Computer Vision and AI Research Group. She enjoys excellent collaboration with national and international institutions and industry. She is the principal investigator of several funded research projects generating approximately five million. She has published extensively in the field of pattern recognition and delivered workshops on teaching programming for undergraduate students. She attended workshops related to higher education strategy, assessment methods, and interactive teaching. In 2015, she was elected as the IEEE Chair for Qatar Section.

AHMED BOURIDANE (Senior Member, IEEE) received an ''Ingenieur d'Etat'' degree in electronics from ''Ecole Nationale Polytechnique'' of Algiers (ENPA), Algeria, in 1982, an M.Phil. degree in electrical engineering (VLSI design for signal processing) from the University of Newcastle-Upon-Tyne, U.K., in 1988, and an Ph.D. degree in electrical engineering (computer vision) from the University of Nottingham, U.K., in 1992. From 1992 to 1994, he worked as a Research Developer in telesurveillance and access control applications. In 1994, he joined Queen's University Belfast, Belfast, U.K., initially as Lecturer in computer architecture and image processing and later on he was promoted to Reader in Computer Science. In 2009, he joined Northumbria University at Newcastle leading the Computational Intelligence and Visual Computing Lab. His is now Professor of Machine Intelligence and Director of Cybersecurity and Data Analytics Research Center at the University of Sharjah, UAE. His research interests are in machine learning with applications to imaging for forensics and security, quantitative pathology and bi-medical engineering, homeland security and video analytics. He has authored and co-authored more than 350 publications and two research books on imaging for forensics and security; and Biometric Security and privacy.