An Analysis and Implementation of a Deep Learning Model for Image Steganography

Raksha Ramakotti(✉) and Surekha Paneerselvam

Department of Electrical and Electronics Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India
bl.en.p2ebs19009@bl.students.amrita.edu, p_surekha@blr.amrita.edu

Abstract. Steganography is the technique that involves hiding a secret data in an appropriate carrier. The major challenge involved in steganography is to ensure that the hidden data does not attract any attention towards it and hence works under the assumption that if the secret feature is visible, then the point of attack is evident. In this work, a novel deep learning model is designed to perform digital image steganography. The dataset used to train the model is Common Object in Context (COCO). An analysis is conducted based on batch size hyper-parameter, to evaluate the performance of the model. Also, the effect of using grayscale and color images on the evaluation metrics of the model is estimated. The analysis was orchestrated by evaluating the average Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) of the trained images. The analysis has produced state-of-the-art results with optimized parametric values and has boosted computational efficiency producing a promising architecture to perform steganography.

Keywords: Image steganography · Convolutional neural networks · COCO dataset · Batch size · Peak signal to noise ratio · Structural similarity index

1 Introduction

Steganography is the art of covered or hidden writing where the goal is to covertly communicate a digital message. The word steganography is derived from two Greek words, steganós, meaning "covered or concealed", and graphia, meaning "writing" [1]. The main goal here is to communicate securely in a completely undetectable manner and to avoid drawing suspicion to the transmission of a hidden data. Hence, the nature of the information format and its quantity plays an important role. The data can be hidden in basic formats like: audio, video, text and images [2]. However, images are considered to be standard carriers because a text message cannot hide bulky data, audio is sensitive to noise and video steganography involves extensive pre-processing and as a data format per se, it is too heavy to analyse [3].

N. Kumar et al. (Eds.): UBICNET 2021, LNICST 383, pp. 210–224, 2021.
https://doi.org/10.1007/978-3-030-79276-3_16

The revolution in digital information has proved the need to send the message in a safe manner. Primarily, three such techniques have come into existence, i.e., cryptography, watermarking and steganography [4,5]. Though cryptography and watermarking are widely used, they showcase a few limitations. Regardless how strong is the cryptographic- encryption algorithm, it provides a scope of being decoded. In case of watermarking, the capacity of information chosen is limited by the application. It is steganography that potentially and effectively bridges these gaps. The advantage of steganography when compared to other methods is that the trace of secret information is unknown [6]. Media files are ideal for steganographic transmission because of their large size. The change is so subtle that someone who is not specifically looking for it is unlikely to notice the alteration.

In the previous work many image processing techniques have been used to implement the concept of steganography. Broadly, these methods can be categorised into three namely spatial domain, transform domain and machine learning techniques. Spatial domain aims to represent a grayscale image as a 2-D matrix or a color image as a 3-D vector of 2-D matrices. Least significant bit (LSB) steganography is a primary and fundamental method that works by replacing the LSBs of selected pixels in the cover image with secret message bits. Other popular spatial domain techniques include Edge Based Data Embedding and Random Pixel Embedding Methods [7,8]. The major transform domain techniques imparted for steganography include Discrete Cosine Transform (DCT) [9], Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT). Setiadi et al. have proposed an algorithm that conglomerates DCT alongside with OTP encryption [10]. Another novel ensemble technique that combines DCT and DWT are proposed in [11,12]. To implement image steganography, the DFT coefficients are modulated such that the secret information can be safely stored in a cover image. At the encoder end, the modulation presents the steganographic image. At the decoder side, this image is decomposed into frequency elements. In [13], Mandal proposes a DFT based image steganographic algorithm with a capacity of embedding payload of 0.75 bpB.

Machine Learning algorithms are mathematical models based on a set of training data which are used in the decision making process without a distinct instruction. Literature survey suggests that these models are broadly used in classification and regression [14,15]. Deep learning is a subset of Machine Learning that mimics the workings of the human brain in processing data for use in detecting objects, recognizing speech, and making decisions, to name a few [16–19]. The edge presented by Deep Learning is that it evinces the ability to learn without human supervision, drawing from data that is both unstructured and unlabelled. Convolutional Neural Network (CNN) is a special type of neural network that is specially designed to implicitly understand the intrinsic properties of images. With input being images, these networks are trained to perform a particular functionality to obtain the desired output. The handshake of neural networks to perform steganography has proven to improve the performance, robustness and efficiency of secret image communication. In [20], Baluja presents a robust model to perform color-color secret-cover image steganography trained using ImageNet dataset and evaluated his model obtaining Squared Summed Error values of cover and secret images using StegExpose Tool. The authors in [21,22], portray neural network models, emphasising on the fact that using grayscale images reduce the payload of the secret image on the cover image while hiding the data. In [23], the authors present a comparative study on LSB substitution technique and CNN based architecture. Wu et al.'s whole processing pipeline consists of two almost identical neural network structures responsible for encoding and decoding.

In this work, a deep learning model has been analysed and implemented to perform image steganography using CNN. The model is trained using COCO dataset and is analysed using Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Section 2 elucidates the methodology and presents a brief description on CNN and the dataset. Furthermore, Sect. 3 unfurls the discussion of analysis and results. The paper culminates with Sect. 4 which talks about the conclusion and future scope of this work.

2 Methodology

In this section, a brief description on CNN and the dataset used is provided. Along with this, proposed architecture for steganography is also presented.

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [20,21,23] are deep learning models that take a series of input images, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one image from the other. Unlike the primitive image methods, the filters and characteristics are not hand-engineered, instead the model is trained to learn and adapt these features to perform the desired functionality. The architecture portrayed by CNN is analogous to that of the connectivity design of neurons in the human brain. CNNs are designed to capture the spatial and temporal dependencies in an image through the application of relevant filters. These models are trained to understand the complexity and sophistication involved in images.

While designing an end to end encoder-decoder CNN model that performs steganography, the following properties are to be noted [24]:

- High Capacity: Amount of information can be embedded into an image.
- Perceptual Transparency: The property of a steganographic image that ensures the quality of the statistical properties of secret and cover image.
- Robustness: After embedding, data should stay intact if stego-image goes into some transformation such as cropping, scaling, filtering and addition of noise.
- Computation Complexity: The measure of computational expensiveness for designing a steganographic model.
- Temper Resistance: The degree with which the secret image is prone to being externally attacked.

2.2 Generalized Architecture for Steganography

The general architecture for steganography using CNN is presented in Fig. 1. Each of the blocks are further explained in the following subsections.

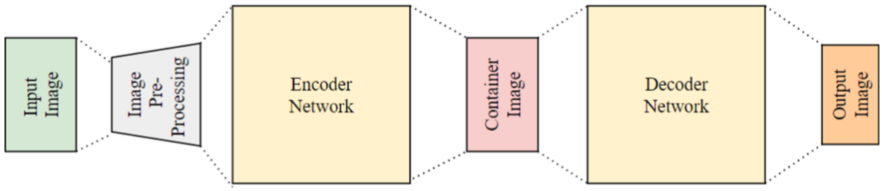

Fig. 1. Block diagram of a general architecture for steganography

Input Image. The input to the steganographic model is a raw image dataset. These images are digital in nature as they can be expressed in terms of a finite set of digital values. The images can be RGB (color) images or Grayscale in nature. Each image is described in (m × n × p) format, where (m × n) represents the size of the image and p represents the number of color channels. Therefore, it takes the value of 1 in case of grayscale image and 3 for a color image.

The smallest element of an image is called a pixel, which in case of color models take the value between 0–255. Thus, the color channels can be represented as an array of 8 bits. This shows that the image can be represented as a function,

$$b = f(m, n) \tag{1}$$

where the value b represents the pixel coordinate value at that particular point.

Dataset Description. COCO stands for Common Object in Context. It is simply a conglomeration of everyday objects captured from everyday scenes. This adds some "context" to the objects captured in the scenes. COCO provides multi-object labeling, segmentation mask annotations, image captioning, key-point detection and panoptic segmentation annotations with a total of 81 categories, making it a very versatile, flexible and multi-purpose dataset. This dataset being open-source was particularly chosen as it introduces the concept of generalization amongst images through non-iconic images [25]. In this work, a subset of 2000 images were used to conduct the analysis.

Image Preprocessing. Image preprocessing is one of the most fundamental and important processes in any image related problem statement. Unlike the conventional digital image processing techniques, where preprocessing of the image is done through a set of algorithms and mathematical calculations, the construction of CNNs removes these computational and mathematical complexities involved. The outlook of preprocessing presented here is in the context of steganography.

Image Reading. The dataset is identified from its storage location path. The raw images are then converted into referable arrays of image matrices.

Image Resizing. The images in the dataset may have different sizes. However, the neural networks demand a base size of an image to perform the computations. Hence the establishment of a uniform size plays a significant role in image preprocessing.

Image Normalization. In the context of image processing, normalization refers to changing the range of the pixel values. An image I can be represented as

$$I = \sum_{i=0}^{M}\sum_{j=0}^{N} P(i,j) \tag{2}$$

here M and N are equal to 255 and P represents the pixel value at the i^{th} and j^{th} coordinate in the image surface.

Since the range is too big, having this representation for the pixel size may increase mathematical complexity, computational time, increase the storage space and reduce the working efficiency of the model. However, while designing the models, the data scientist must ensure that the cost is not too expensive. Hence image normalization is performed on the same image such that the features of the image and its constructional sophistication is not lost. Thus, Eq. 2 can be reframed as

$$I_{norm} = \sum_{i=0}^{1}\sum_{j=0}^{1} P(i,j) \tag{3}$$

Dataset Splitting As a part of preprocessing technique that is specifically pertaining to training a steganographic model, the images are split into secret and carrier images. In some of the test cases, the secret images are converted to grayscale in this step.

Steganographic Model Description. The end-to-end process of steganography as showcased in Fig. 2 can be split into two:

Encoder. This process has two important purposes each of which is performed by two neural networks:

- Composition Network: This network takes the secret image as its input image. The purpose of having a dedicated network for a secret image is to understand the composition of the image. This network is trained to extract the features of the secret image and present it to the Conceal Network in such a way that there is slight to nil trace of the secret image on the carrier image. The output of the Composition Network is a transformed secret image which is fed to the Conceal Network.

- Conceal Network: It is in this network where the actual functionality of steganography is performed. The input to the Conceal Network consists of two things, one the transformed secret image and the other is the cover image. The model is to be trained in such a way that the statistical property of the cover and or the secret image is not distorted in the process. Since images provide a high capacity to hide another message, the trade-off between information heaviness and transparency must be taken care of ensuring that the very purpose of steganography is fulfilled. The output of this network is a container image. The model should be accurately trained in such a way that the container image looks similar to that of the carrier image.

Decoder. The process of revealing the secret image from the container image is known as steganalysis. This is done by the decoder which consists of the Exhibit Network. The presence of this network is what makes the complete process of steganography to be comprehensive and complete. The task of the Exhibit Network is to remove the carrier image without distorting the structural integrity of the secret image. Strikingly, the container image is the input to the Exhibit Network and the output image is to be as similar as the secret image.

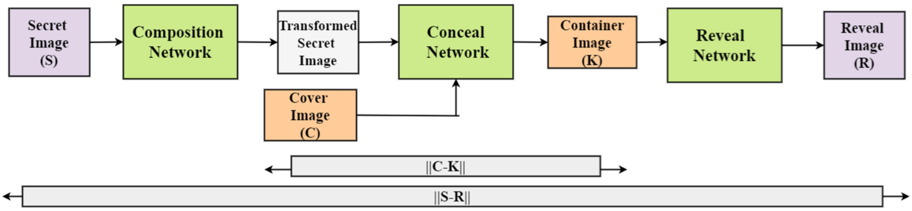

Fig. 2. End to end steganographic model architecture

Proposed Neural Network Architecture. The model architecture as described in Fig. 2 includes Composition, Conceal and Exhibit Networks. As shown in the figure, these networks are concatenated and trained together. The model's structure is primarily designed based on defining certain variables called hyper-parameters which are tabulated in Table 1.

The complete detailed model is presented in Fig. 3 Though the architecture borrows the structure from Baluja's model [20], there are some distinct differences. One, the kernel size of uniform (3×3) is utilized instead of (3×3), (4×4) and (5×5) because:

- Using even sized kernels does not produce a computational simplicity as all the previous layer pixels would not be symmetrically around the output pixel. This would lead to distortions as the center pixel cannot be interpolated.

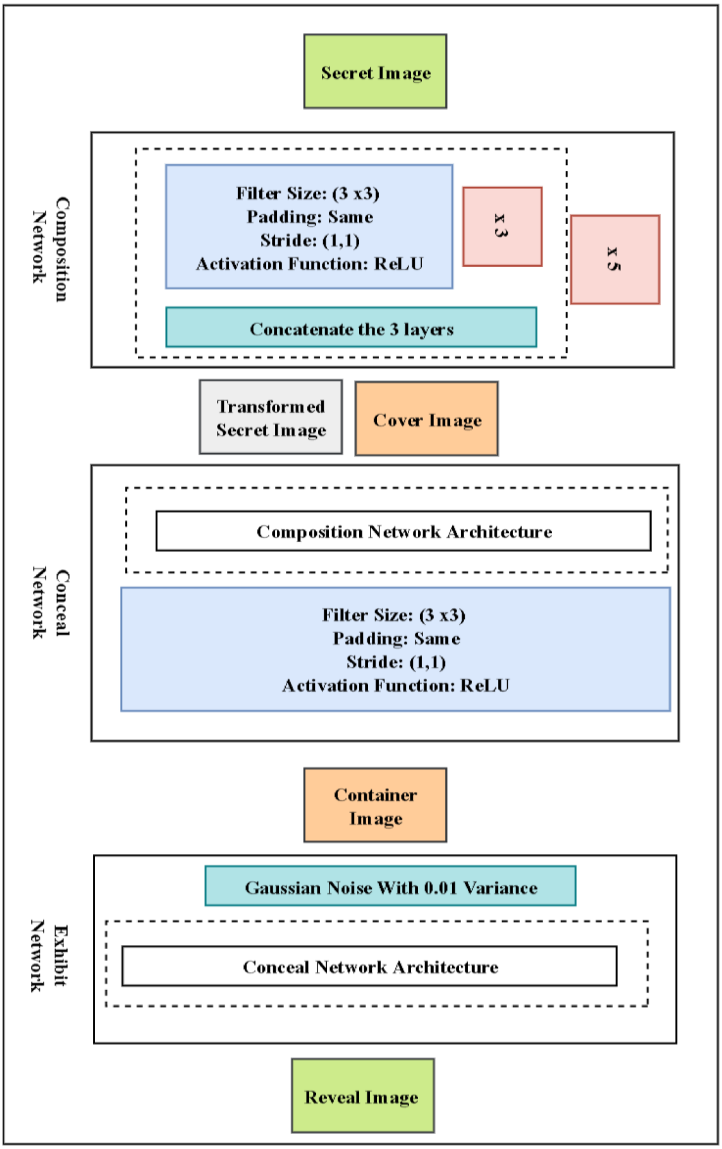

Fig. 3. Detailed depiction of proposed model architecture

- Using large kernel sizes such as (5 $\times$ 5) would increase the number of weights to be back-propagated which would lead to long training time and using more computational resources to do so.

Table 1. Hyperparameter tabulation for the proposed work

Hyper-parameter	Value
Number of layers	60 (20 in each network)
Strides	(1, 1)
Padding	Same
Kernel size	(3 × 3) in all layers
Activation function	ReLU
Epochs	100
Batch size	10–50
Learning rate	0.001

The kernel in a CNN model is a weighted matrix that is used to extract certain features of the image. This kernel is slid across the input image and convolution operation is performed such that the desired operation for which the network is designed is successfully performed.

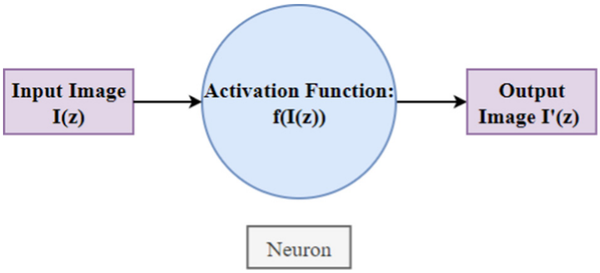

Fig. 4. Depiction of the working of activation function

The number of kernels in each layer is defined by the number of neurons. Each unit consists of a fixed number of neurons. In Baluja's model, it is described that each layer in each unit consists of 50 neurons. However, in the scope of proposing a low-parameter neural network, the architecture of the model here is constructed differently. The layers of the first unit have 50 neurons each. With each passing unit 10 neurons are reduced in order to afford the computational cost. The layers of the last unit thereby consist of 10 neurons each.

As shown in Fig. 4, the filtered image is passed to a mathematical "gateway" known as the activation function. It aids in transforming the input into a desired output through the mathematical definition of the function. The activation function used is ReLU which stands for Rectified Linear Unit. As shown in Fig. 5 it is a nonlinear piece-wise function, where the negative inputs are ceiled to 0 and for 0 and other positive inputs obey the mathematical formula,

$$b = f(0, n) \tag{4}$$

where n belongs to positive integers.

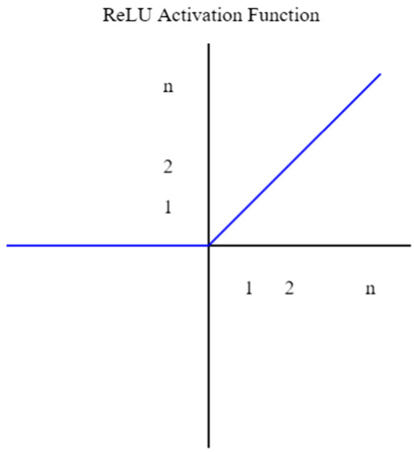

Fig. 5. Graph of ReLU Activation Function

As depicted in Fig. 2 error term i.e., the absolute difference of C and K, showcases the difference between cover and container image and it back propagates only through the first two networks and the error term representing the absolute difference of S and R,is calculated by taking the difference between secret and the reveal image. This term back propagates through all the three networks. Thereby, the entire loss equation that is thus used to train the model is given by,

$$L(C, K, S, R) = ||C - K|| + ||S - R|| \tag{5}$$

3 Analysis and Results

The evaluation metrics used to estimate the performance of the model analysed model architecture is presented in this section. COCO dataset was analyzed for cases where the secret images were color and grayscale. All the models were implemented in python 3 script in Google Colab. Tensorflow framework is used to model CNNs.

3.1 Evaluation Metrics

In Image Processing and Computer Vision applications, where both the input and the output are digital images, the quantifying evaluation metrics used to determine the performance of the are PSNR and SSIM. The details are provided in the following subsections.

Peak Signal to Noise Ratio. The Peak Signal to Noise Ratio (PSNR) [26] represents the ratio between the maximum power value of a signal and the power of distorting noise that affects the quality of its representation. The higher the PSNR, the better the quality of the reconstructed image. It is measured in decibels. The formula for PSNR is given as,

$$PSNR = 10log_{10}(\frac{MAX_{y_{ij}}^2}{MSE}) \tag{6}$$

where, MSE stands for Mean Squared Error and the numerator term represents the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

Structural Similarity Index. The Structural Similarity Index (SSIM) [27] is an enduring metric that quantifies the degradation of an image quality due to extensive image processing or modelling. SSIM is given by,

$$SSIM(I, O) = (\frac{(2m_I m_O + c_1)(2s_{IO} + c_2)}{(m_I^2 + m_O^2 + c_1)(s_I^2 + s_O^2 + c_2)}) \tag{7}$$

where,
· m_I m represents the average of the input image I
· $m_O$represents the average of the output image O
· s_I represents the variance of the input image O
· s_O represents the variance of the output image O
· s_{IO} represents the covariance of the images I and O

It is to be noted that the evaluation metrics were calculated separately between the Cover and Container Images at the encoder end and Secret and Reveal Images at the decoder end. In each of these cases, Cover and Secret Images will be treated as the input image I and the corresponding Container and Reveal Images as the output image O, which is represented in the Eq. 7.

3.2 Performance Analysis of the Steganographic Model

Image Dataset: Cover and Secret Images Are Color. The model is constructed with the ReLU activation function in all the layers. The batch sizes are varied between 10 and 50. Figure 6. shows the variation of Encoder and Decoder Networks' PSNR with respect to batch size for COCO dataset, where both the cover and secret images are color images.

It can be observed that the encoder model showed an almost consistent performance with respect to PSNR, with its values in the range 26.91 dB and 27.18 dB. The decoder however showcased a slight descending drop in the PSNR value, when the batch size was increased. At batch size = 30, both the encoder and the decoder network an approximate value of 27 dB. The average PSNR value showcased by both the networks were quite similar too, i.e., 27.02 dB and 26.98 dB. However, at batch size =10, both encoder and decoder value gave the best result,

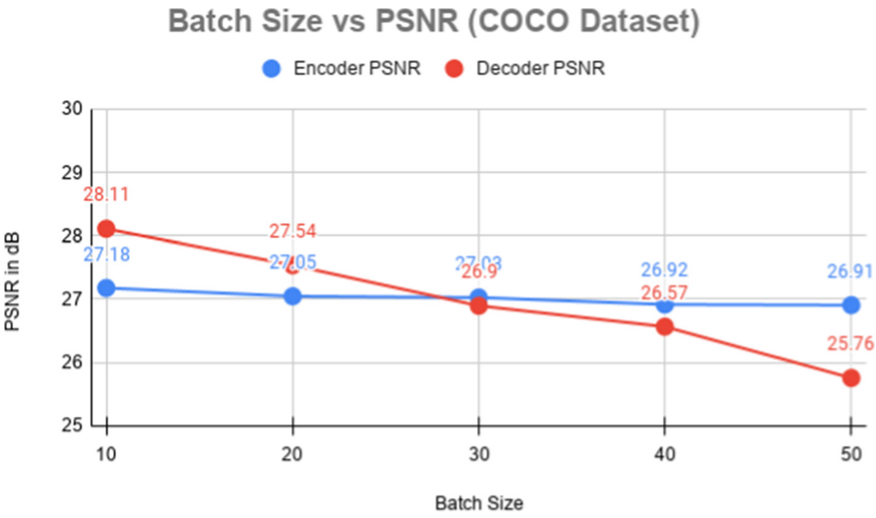

Fig. 6. Performance analysis of COCO dataset PSNR with respect to batch size

Table 2. Evaluation metrics tabulation of COCO dataset when cover and secret images are color

Batch size	Encoder PSNR (dB)	Decoder PSNR (dB)	Encoder SSIM	Decoder SSIM
10	27.18	28.11	0.94	0.93
20	27.05	27.54	0.94	0.93
30	27.03	26.90	0.94	0.93
40	26.92	26.57	0.94	0.93
50	26.91	25.76	0.94	0.93

i.e., 27.18 dB and 28.11 dB respectively. Since the COCO dataset introduced the characteristic of generalization, the stability in the PSNR values could be visibly observed from the graph. Interestingly, this consistency can be observed with the SSIM parameter as well. For all the batch sizes, the model reflected an SSIM of 0.94 and 0.93 for the container image and reconstructed image respectively. These values are tabulated in Table 2.

Image Dataset: Cover Images Are Color and Secret Images Are Grayscale. The same architecture showcased in Fig. 3 was trained for a scenario where the secret image was grayscale. The values of the evaluation metrics are tabulated in Table 3.

The trade-off between the encoder and the decoder performance can evidently be observed here. As portrayed in Fig. 7, the encoder PSNR drops when the secret image is grayscale. This is because the encoder network tends to treat the transformed grayscale image as a seemingly noise component, rather than the signal itself. The COCO dataset heavily suffers on this criteria. The average performance of the PSNR encoder for all the batch sizes was calculated to be

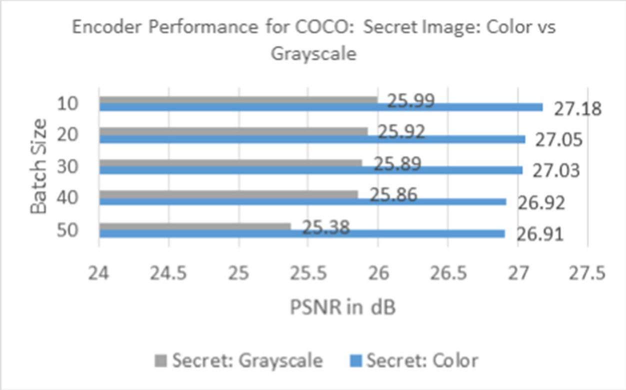

Fig. 7. Performance analysis of encoder with respect to batch size for color vs grayscale secret images for COCO dataset

Table 3. Evaluation metrics tabulation of COCO dataset when cover images are color and secret images are grayscale

Batch size	Encoder PSNR (dB)	Decoder PSNR (dB)	Encoder SSIM	Decoder SSIM
10	25.99	34.6	0.93	0.98
20	25.92	33.06	0.93	0.98
30	25.89	29.4	0.93	0.98
40	25.86	28.82	0.93	0.98
50	25.38	28.54	0.93	0.97

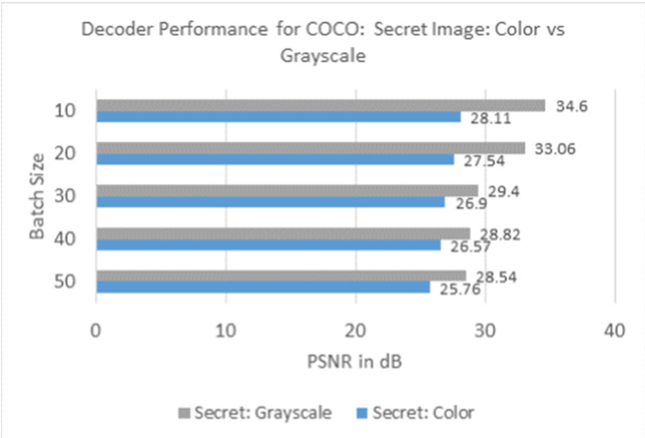

Fig. 8. Performance analysis of decoder with respect to batch size for color vs grayscale Secret images for COCO dataset

25.81 dB, which has dropped by 4.48% when compared to that of the average performance of PSNR encoders when color images are used as secret images.

The decoder performance on a contrary performed better with the secret image being grayscale in nature. The reason could be because of the reduction in the payload by, i.e., because of eliminating a 3 channeled image to expose a 1 channeled grayscale image. With the reduction in the batch size, the performance was improved. It is however with the use of COCO dataset, it could be understood the working of the Exhibit Network. Since the grayscale image is treated as a noise, the decoder network promptly tries to adapt the working of a denoising model. The batch sizes 20 and 10 evaluated excellent PSNR results, i.e., 33.06 dB and 34.6 dB respectively. This sufficiently proves the effect of reduction of the payload by 66.66% directly impacts the working of the decoder model. Figure 8 shows the performance of the decoder for batch sizes 10 to 50.

4 Conclusion and Future Scope

The implementation of a novel model for steganography has been analyzed successfully. The proposed model was designed and trained with the COCO dataset, which was used to induce the property of generalization. It was observed that for a batch size of 10, the models performed the best. COCO was consistent with its PSNR and SSIM values i.e., around 27 dB and 0.935 respectively. When the secret images were converted to grayscale, there was a dip in the encoder performance as the model treated the transformed secret images as noise components. On the contrary, with the reduction in the payload on the decoder end, "denoising" functionality proved to be an efficient task. The current model can be improved by increasing computational efficiency, training the model with more iterations, increasing the number of training images and further tuning the hyper-parameters of the model. Also, the model can be condensed to fit into the needs of being called a 'TinyML' model, where the complex process of steganography is implemented by an Embedded System processor.

References

1. Pund-Dange, S.: Steganography: a survey. In: Bokhari, M.U., Agrawal, N., Saini, D. (eds.) Cyber Security. AISC, vol. 729, pp. 327–333. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8536-9_32
2. Febryan, A., Purboyo, T.W., Saputra, R.E.: Steganography methods on text, audio, image and video: a survey. Int. J. Appl. Eng. Res. **12**(21), 10485–10490 (2017)
3. Menon, N.: A survey on image steganography. In: 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), pp. 1–5. IEEE, December 2017
4. Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P.: Digital image steganography: survey and analysis of current methods. Signal Process. **90**(3), 727–752 (2010)
5. Al-Yousuf, F.Q.A., Din, R.: Review on secured data capabilities of cryptography, steganography, and watermarking domain. Indonesian J. Electr. Eng. Comput. Sci. (IJEECS) **17**(2), 1053–1059 (2020)

6. Pillai, B., Mounika, M., Rao, P.J., Sriram, P.: Image steganography method using k-means clustering and encryption techniques. In: 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1206–1211. IEEE, September 2016
7. Hussain, M., Wahab, A.W.A., Idris, Y.I.B., Ho, A.T., Jung, K.H.: Image steganography in spatial domain: a survey. Signal Process.: Image Commun. **65**, 46–66 (2018)
8. Wang, Z.H., Chang, C.C., Li, M.C.: Optimizing least-significant-bit substitution using cat swarm optimization strategy. Inf. Sci. **192**, 98–108 (2012)
9. Kadhim, I.J., Premaratne, P., Vial, P.J., Halloran, B.: Comprehensive survey of image steganography: techniques, evaluations, and trends in future research. Neurocomputing **335**, 299–326 (2019)
10. Rachmawanto, E.H., Sari, C.A.: Secure image steganography algorithm based on DCT with OTP encryption. J. Appl. Intell. Syst. **2**(1), 1–11 (2017)
11. Sari, W.S., Rachmawanto, E.H., Sari, C.A.: A good performance OTP encryption image based on DCT-DWT steganography. Telkomnika **15**(4), 1987–1995 (2017)
12. Kumar, V., Kumar, D.: Digital image steganography based on combination of DCT and DWT. In: Das, V.V., Vijaykumar, R. (eds.) ICT 2010. CCIS, vol. 101, pp. 596–601. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15766-0_102
13. Mandal, J.K., Khamrui, A.: A genetic algorithm based steganography in frequency domain (GASFD). In: 2011 International Conference on Communication and Industrial Application, Kolkata, India, pp. 1–4 (2011)
14. Raksha, R., Surekha, P.: A cohesive farm monitoring and wild animal warning prototype system using IoT and machine learning. In: 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 472–476. IEEE, October 2020
15. Se, S., Vinayakumar, R., Kumar, M.A., Soman, K.P.: Predicting the sentimental reviews in Tamil movie using machine learning algorithms. Indian J. Sci. Technol. **9**(45), 1–5 (2016)
16. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. **30**(11), 3212–3232 (2019)
17. Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A.E.D., Jin, W., Schuller, B.: Deep learning for environmentally robust speech recognition: an overview of recent developments. ACM Trans. Intell. Syst. Technol (TIST). **9**(5), 1–28 (2018)
18. Ganesan, S., Krichen, M., Alroobaea, R., KP, S.: Robust Malware Detection using Residual Attention Network (2020)
19. Poorna, S.S., et al.: Computer vision aided study for melanoma detection: a deep learning versus conventional supervised learning approach. In: Pati, B., Panigrahi, C.R., Buyya, R., Li, K.-C. (eds.) Advanced Computing and Intelligent Engineering. AISC, vol. 1082, pp. 75–83. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-1081-6_7
20. Baluja, S.: Hiding images in plain sight: deep steganography. In: Advances in Neural Information Processing Systems, pp. 2069–2079 2017)
21. Rahim, R., Nadeem, S.: End-to-end trained CNN encoder-decoder networks for image steganography. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
22. Zhang, R., Dong, S., Liu, J.: Invisible steganography via generative adversarial networks. Multimed. Tools Appl **78**(7), 8559–8575 (2018). https://doi.org/10.1007/s11042-018-6951-z
23. Wu, P., Yang, Y., Li, X.: Stegnet: mega image steganography capacity with deep convolutional network. Future Internet **10**(6), 54 (2018)

24. Pandikumar, T., Gebreslassie, T.: Information security using image based steganography. Int. Res. J. Eng. Technol. (IRJET) **3**(06), 2839–2844 (2016)
25. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
26. Hore, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 2010 20th International Conference on Pattern Recognition, pp. 2366–2369. IEEE, August 2010
27. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)