Advanced Intelligent Data Hiding Using Video Stego and Convolutional Neural Networks

Eman S. Harba[1]   **Hind S. Harba[2]**   **Inas Ali Abdulmunem[3*]**

[1]Avicenna Unit for E-Learning, College of Arts, University of Baghdad, Baghdad, Iraq
[2]Department of Atmospheric Sciences, College of Science, University of Mustansiriyah, Baghdad, Iraq
[3]Computer Science Department, College of Science, University of Baghdad, Baghdad, Iraq
[*]Corresponding author: emanharba_121212@coart.uobaghdad.edu.iq, hindharba11.atmsc@uomustansiriyah.edu.iq, inas.ali@.uobaghdad.edu.iq[*]
[*]ORCID ID: https://orcid.org/0000-0002-6407-9171, https://orcid.org/0000-0002-1980-2578, https://orcid.org/0000-0002-8813-7524[*]

Abstract:

Steganography is a technique of concealing secret data within other quotidian files of the same or different types. Hiding data has been essential to digital information security. This work aims to design a stego method that can effectively hide a message inside the images of the video file. In this work, a video steganography model has been proposed through training a model to hiding video (or images) within another video using convolutional neural networks (CNN). By using a CNN in this approach, two main goals can be achieved for any steganographic methods which are, increasing security (hardness to observed and broken by used steganalysis program), this was achieved in this work as the weights and architecture are randomized. Thus, the exact way by which the network will hide the information is unable to be known to anyone who does not have the weights. The second goal is to increase hiding capacity, which has been achieved by using CNN as a strategy to make decisions to determine the best areas that are redundant and, as a result, gain more size to be hidden. Furthermore, In the proposed model, CNN is concurrently trained to generate the revealing and hiding processes, and it is designed to work as a pair mainly. This model has a good strategy for the patterns of images, which assists to make decisions to determine which is the parts of the cover image should be redundant, as well as more pixels are hidden there. The CNN implementation can be done by using Keras, along with tensor flow backend. In addition, random RGB images from the "ImageNet dataset" have been used for training the proposed model (About 45000 images of size (256x256)). The proposed model has been trained by CNN using random images taken from the database of ImageNet and can work on images taken from a wide range of sources. By saving space on an image by removing redundant areas, the quantity of hidden data can be raised (improve capacity). Since the weights and model architecture are randomized, the actual method in which the network will hide the data can't be known to anyone who does not have the weights. Furthermore, additional block-shuffling is incorporated as an encryption method to improved security; also, the image enhancement methods are used to improving the output quality. From results, the proposed method has achieved high-security level, high embedding capacity. In addition, the result approves that the system achieves good results in visibility and attacks, in which the proposed method successfully tricks observer and the steganalysis program.

Key words: Convolutional neural networks, Hiding Data, Image Stego, Steganography, Video Stego.

Introduction:

Steganography comes from Greek's two words: steganos, which means "protected" or "covered" and "graphs," which means "writing." Steganography is the technology of communicating in such way the existence of a secret message can be hidden, or in other words, in a simple "it is a type of a security through obscurity". Strategies to hide secret information is used for centuries, however, recently the most of information is transferred using an electronic format; thus, there is a tremendous growth of algorithms in both of steganography and watermarking techniques for concealing importantly

(secret) information within an irremovable (secure) and undetectable way in mainly of 4 types; image,text, audio/video and protocol which are well explained in (1). Steganography is often combined with cryptography to improve the security of the hidden message. Steganography is commonly used in applications such as digital watermarking, private communication, secret data storing, etc. (1, 2).

Images that were used for hiding secure information are called 'cover images,' and the image where secret bits are inserted is called stego image. Steganography on images can be broadly classified as frequency domain and spatial domain steganography (3). In the frequency domain, there is a change in some mid-frequency parts, while in the spatial domain, the algorithm is directly changing the values (least significant bits) for some selected pixels. These heuristics are efficent in the areas for which they are designed (hiding), but they are essentially static and therefore easily detectable. A steganographic technique or algorithm can be evaluated through the use of performance and quality metrics such as robustness, capacity, applicability, imperceptibility, etc. (4). There are several publicized nefarious approaches for data hiding using steganographic techniques, like coordinating and planning criminal activities through hide messages within an image in public site causing the conversation and the recipient is difficult to observe (5). The task of strong steganography arises due to the embedding of the secret message that can alter the style and underlying carrier statistics. The quantity of alteration depends upon two factors: the first factor is the quantity of data that hasbeen hidden. The second factor is that the quantity of alteration depends upon the carrier image itself. Most of the steganography algorithms are based on the modification of the pixels or mathematical equations that are applied to the images before embedding. Based on this, the embedded process is perform using spatial domain methods and transform domain (6).

New methods which are used to hide data (such as images or video) in video file already exist; however, there are many issues relating to these methods. The first and most impotent issue that these methods are not hard to decode because the encoding strategy for information is fixed. The second issue is that the amount of information (Data) that can be hidden is often less; for example, when hiding an image or video within the same size/resolution of image or video, it will probably cause losing a fair bit of information. The third issue, in the case of images, the traditional stego algorithms do not exploit the structure of the images, because they do not make use of the patterns found in original images (7). In this work, a solution based Neural Network has been proposed to overcome these issues. Convolutional neural networks (called CNN) have been shown to learn structures that correspond to logical features. These kinds of features raise their level of abstraction when they go deeper through the network. By using a CNN in this approach, most of the issues mentioned previously are solved. The first advantage, is that the convent would is a good idea with regards to patterns of original images and are going to be capable of making decisions on what areas are redundant. As a result, more pixels will be hidden there (8). Through saving space on redundant areas, the hidden information amount will be increased since the weights and architecture could be randomized. As a result, the exact way by which the network will hide the information is unable to be known to anyone who does not have the weights. A Convolutional Neural Networks (CNNs) is a type of Deep Learning algorithm that can take an input images, identify importance (biases and learnable weights) to several aspects/objects in the image {and have the ability to recognize one from the others. The training protocol is comparable to traditional machine learning techniques. Eavery image is given as input for the network. Every pixel value is transferred to several neurons. The network involves many blocks (9, 10). The network involves many blocks. These blocks involve neurons that take actual input values, carry out calculations, and send the real calculated values to another block. A neural networks can, consequently, be symbolized by an oriented graph in which every node represents a computing element. The training is then completed by providing the network with examples consists of an image, and it is the label. The network changes the parameters of the calculation units (i.e it learning) because of the process of back-propagation (11). The CNN utilized for steganography and steganalysis are generally built-in from three stages, which are: the preprocessing stage, the convolution stage, and the classification stage. As an illustration in Fig. 1 shows the network suggested by the Yedroudj-Net Network (12, 13), in which the network process the gray-scale images of 256×256 pixels.
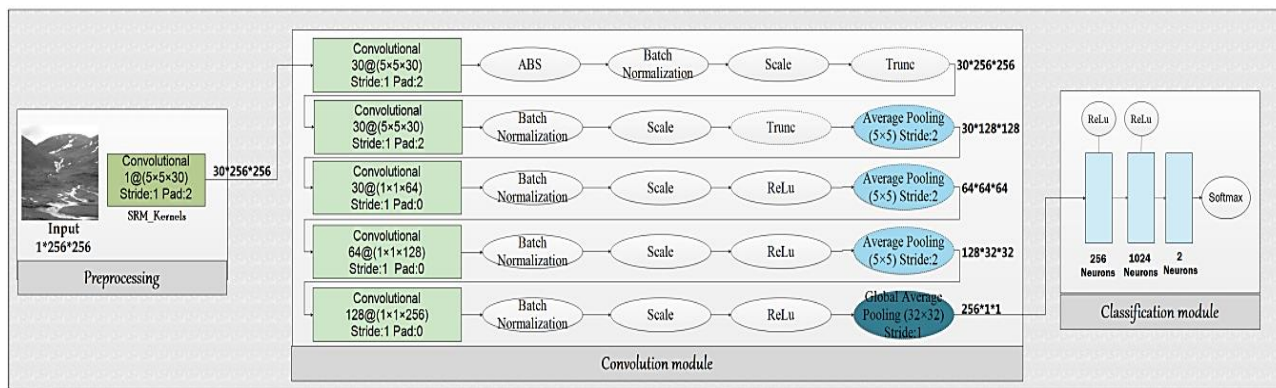
Figure 1. Yedroudj-Net network (12).

With regard to the latest impressive results achieved work is done by (14), where they used CNN with steganalysis, there are relatively little researches to add neural networks for the hiding process itself. Several of these researches utilized CNN to choose what is the best LSBs to substitution an image with the binary to embed a text message. Others, (15-17), have utilized CNN to determine which is the desired bits to extract from the cover images. In this work, the neural network is used to determines where it can place the secret data and how exactly to encode it effectively; the hidden message can be disappeared through the entire bits in the image. On the other hand, the decoder network is trained simultaneously with the encoder so it can be used to reveal the secret message. Taking into consideration that these networks are trained just one time, and they are independent of both secret or cover images. In this study, the essential implementation from the task (14) is intend to extend to propose video steganography, where the model have trained in order to hiding videos (secret message) inside another video (cover) by using CNN.

Related Works

Steganography has a very long history of evolution. Methods in early steganography are elementary: the secret messages are hidden within another form. These methods are essential, and therefore, it may be easily detected. For that reason, high secure steganography approaches are required. With growing of digital signal processing, a large number of digital steganography methods have been developed. Kurak and McHugh (1992) (18), designed the earliest method of digital steganography approaches. They performed embedding data into four LSB of an image. Concealing secret messages within another form in addition to an image is also possible. Hernandez et al. (2006) (19), were successful in hiding data in XML and HTML documents in addition to an

executable file (.exe). Hosmer (2007) (20), also utilized the LSB method to hide information in the image (JPEG and GIF format) as well as the music file. The steganography based LSB is commonly used; however, it has some main drawbacks. One of the most significant drawbacks of the LSB-based method system is usually that is the lack robustness in the event that facing stego-analysis. To be able to address these issues of the LSB technique, a method employing CNN has been proposed. One of the earliest researches of that field is the work done by Imran Khan et al. (2010) (21), where feature representation could be learned by a neural network automatically that has a number of convolutional layers. In the last years, several researches have been achieved. Qian et al. (2015) (22), proposed a model that is based on utilized CNN, which they called GNCNN. This model used the handcrafted KV filter in order to extract residual noises and applied the Gaussian function to gain more significant features. The performance of the prposed model is lower slightly than the set spatial rich model. Baluja (2017) (14), utilized a NN architecture composed of 3 convolutional networks to hide data in the public images of identical size. Xu et al. (2017) (23), presented the Batch Notmslization to avoid the network falling towards the local minimum. XuNet prepared with Tanh, 1×1 convolution with global average pooling where gained equivalent performance to SRM. Tang et al. (2017) (24), proposed a learning framework for automatic steganographic distortion called ASDLGAN. The proposed model can convert a cover image to an embedding modified probability matrices, and the discriminator includes the XuNet architecture. To be able to fit the optimum embedding simulator along with propagating the gradient for back-propagation, they designed an activation function called ternary embedding simulator (TES). Zeng et al. (2018) (25) developed a JPEG stego-analysis model that has parameters less than the XuNet model and got better accuracy than XuNet. That works that mentioned was applied

CNN to stego-analysis efficiently, but there is yet space for enchantment. Zhu et al. (2018) (26), proposed an end-toend CNN for both encoding/decoding stego images, which can hide secret message as bit string within an image as well as the ability to decoded the encoded image in order to retrieve the original secret message. Zhu has also designed the adversarial network together with a pair of encoder/decoder arrange. This network can determine whether a given image includes some secret message and will help to improve the encoded image quality. Hussain et al. (2018) (27) proposed algorithms dependent on Deep Convolution Neural Network (DCNN). Their method overcomes some limitations on previous techniques related to the limited dataset used in examination and evaluation, in addition, they did not consider external environmental changes. In their work, the images had been directly used as input to DCNN for training and recognition without any extracting features; beyond this, DCNN learns optimum features from images by using the adaptation process. The results show that there proposed approach got the efficient ability of automatically recognizing with a higher accuracy reach to 99%. It is also able to fulfill real-world application criteria efficiently. Meng et al. (2019) (28), proposed a review on image data hiding based upon deep learning. The study divided image data hiding for four parts of algorithms, which are, coverless information hiding, watermarking embedding, steganography, and steganalysis based DNN. From all these aspects, state-ofart data hiding technologies depending on deep learning, are described and analyzed. Duan et al. (2020) (29), proposed a new image steganography approach that achieved high capacity dependent on utilized deep learning. In their work, the Discrete Cosine Transform has been used to convert the secret image. After that, the converted image is encrypted via using Elliptic Curve Cryptography in order to improve the anti-detection feature of the acquired image. Utilizing the SegNet Deep Neural Network in addition to a set of Extraction and Hiding networks to improve steganographic capacity, allows steganography and extract of full-size image. The test results prove that their method can efficiently allocate each pixel inside the image to ensure that it reach to relative capacity (30).

Methodology

The main objective of this work is to hide data (images or video) within another video file, where secret image and the cover image has similar resolution size ((in this work, a multiple secret images of full size (N*N) image of RGB color need to hide). CNN's are trained all at once to generate both hid and reveal processes and are built to work as a pair mainly. The strategy used is image compression using automatic encoding networks. The train system has to learn to compress the data from the secret images into the least visible portions of the cover image. After that, it should learn the best ways to extract and reconstruct the exact data from the encoded message, that have a minimal loss. The architecture of the entire network is the same as Auto Encoders. Generally, automatic encoders are designed to reproduce the input subsequent to a series of conversions. As a result, they learn for the features from the input distribution. The main model consists of three Parts which are: the prepared Network, the revealed Network, and the hidden Network. Fig. 2 shows the Network Architecture.
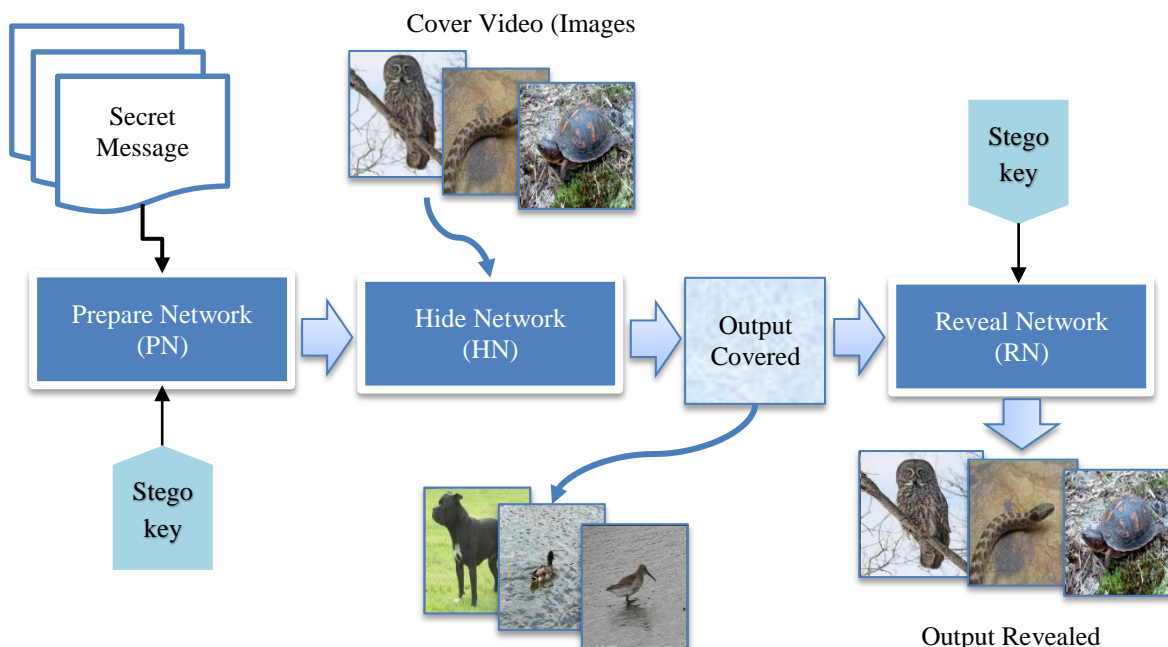
Figure 2. the proposed model architecture

The PN input the secret image to prepares it. The HN takes the PN Output along with the Cover Image. At first, these two inputs are concatenated throughout the Channels Axis. The HN outputs an image, which represents the hidden image. These images include the secret data, but it seems like the cover. To be able to retrieve the original secret image back again, it should be transferred to an RN, which will output an image which is similar to the secret. The real architecture of each of the networks is nearly similar. In this work, a kernel convolutions of (3x3), (4x4), and (5x5) on the input (50 maps) have been used. Then there are three convolutions on the concatenated feature maps. After that, a 1x1 convolution is made to produce three channels.

As a rule, both hiding and revealing networks are trained together in the form of an automatic encoder via using Keras. Our model involves two inputs corresponding to a pairs of secret/cover images and two outputs equivalent to their inputs. An automatic encoder-based architecture was utilizing, and then the labels are exact as their corresponding inputs. The network is composed of three parts viz, which are: prepared, hide, and revealed block. In the first block, the color-based pixels are converted to additional useful features for concisely encoding the images. After that, the transformed image is hidden within the input cover image by using the hidden block to generate the container image.

At last, in the reveal block, the container image then decode to generate the secret output. As a result, the training graph includes two inputs and two outputs. Following (11), a mean square error is used between the actual image pixels and the pixels of the reconstructed image as the metric. As well, a weighted loss function L() besides Adam optimizer are used in order to train the model described in (11).

$$L(B, B_0, A, A_0) = ||B - B_0|| + \beta||A - A_0|| \qquad (1)$$

Here B and A are the covers and the secret images respectively, and β is how to reconstruct errors of their weight. Notably, it can be noted that the error term $||B - B_0||$ is not applied to the decoder network weights. However, the encoder network and the decoder network get the error signal $\beta||A - A_0||$ for rebuilding the secret image. However, the mean square error just got a vast error of two images' related pixels, but it disregards the main structure in images. However, to make sure that the networks will not easily encode the secret image in the LSBs, a little amount of noise is applied to the output of the second network through the training. After the training is completed, the trained model is divided into two parts: hided network and revealed network (and deleted the noise layer).

The HN has two inputs equivalent to secret and cover image in addition to one output corresponding to the cover image. The RN will take the container image as the input and decodes the secret image to get output. In this model, 100 epochs have been used to train the network, and a batch size of 8 have been used. Figure 3 illustrates the proposed network process.

Figure 3. Proposed System Architecture. (a) Hide network, (b) Reveal Network

From Fig. 3, the HN is applied by the sender; and the revealed network is made used by the receiver. The receiver can access just to the container image. Furthermore, the secret images is also encrypt to add extra security. As a result, both the sender and the receiver should share a symmetric key in order to encrypt/decrypt the secret message. On the sender side, the encryption process is done on the secret input image, while the decryption is done on the receiver side to retrieve the secret message.

Results and Discussion:

In this part, the experiment aspects and results are presented. The model is designed by used Python and Conda (by using data science toolkit Anaconda) with some dependent

(TensorFlow, which is the machine learning platform, NVIDIA CUDA Toolkit, Open CV and NVIDIA cuDNN,). The datasets that have been used are large scale computing resources. In our test, an 45000 images from the ImageNet dataset and examined 5000 images have been trained— these images used as secret messages. The cover is a large video file of resolution (1980×1024) and 25 frames per second (fps). The video file is splitting to image per second (15 images per second). All the images that have been used are resized to the resolution of 256×256 pixels without normalization. However, in order to display the visualization of the images during the training and investigate the training progress, a tensor board image logger have been utilized, since the Keras does not have a built-in image logger. The whole process had taken about

17 hours by using one computer of the specification described in Table1.

Table 1. Hardware and software specification used in the test.

CPU	2.2GHz Core i7 intel8750 Bost up to 3.9Ghz
Ram	16GB DDR4
GPU	Nvidia
HDD	1TGB SSD (NVMe)
Operation System	Windows 10 64bit
Software	Python 3

While steganography is sometimes conflated with cryptography, however, in the proposed approach, the nearest analog is image compression throughout auto-encoding network. The proposed model should learn to compress the data from the secret images to the least recognizable portions of the cover images. The image transformation in each network of the proposed system is shown in Fig. 4.


(a)   (b)   (c)
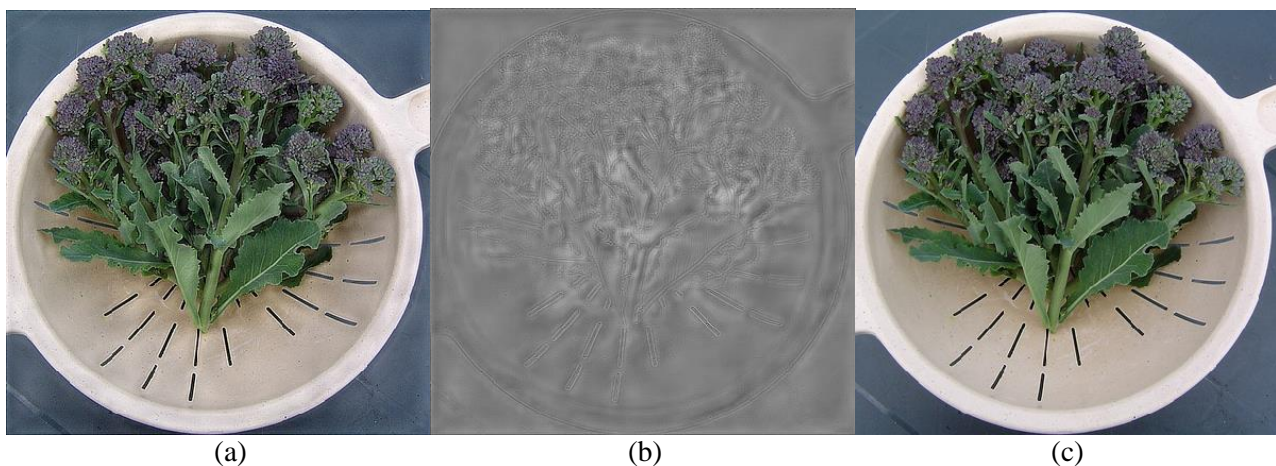
Figure 4. Transformations of an image in the proposed model. (a) Original image in preparation network, (b) the image in hiding Network, (b) the Reveal Network output

The 3 components shown in Fig. 2 had been trained to be as a single network; even so, it is recommended to identify them individually. PN has been prepared the secret image to become hidden. This network assists two purposes.

The first purpose, when the secret-image (of M×M size) is little sized than the cover image of (N×N), then the PN progressively rises the secret image size up to the cover size; therefore the secret image's bits are distributing throughout the entire cover image (N×N) pixels. The second more significant purpose, associated with all hidden images sizes, is to modify the color-based pixels into more useful features to encode the image, like edges. The HN takes the cover image and the output of the PN as input; after that, it creates the Container image. The input of the Network has a size of N×N, and it has a depth concatenated the RGB channels from the covered images and the modified channels from the secret images. Over

thirty architectures of this network have been experimented for our study in addition to a various number of hidden layers and convolution sizes; one of the best has consisted of five convolution layers that have fiften filters where each filter have of 3×3 patches, 4×4 patches and 5×5 patches.

At last, the RN which is used by the receiver, which represents the decoder. The receiver receives only the Container image (which is not the secret nor the cover image). The RN takes out the cover image to retrieve the original secret image. As explained previously, our approach borrows closely from auto-encoding networks, but, rather than simply encoding a single image throughout a bottleneck, a two images have been encoded in a way that the container image (intermediate representation) looks as very similar as is possible to the cover image. The proposed model is trained by minimizing the error (equation 1).
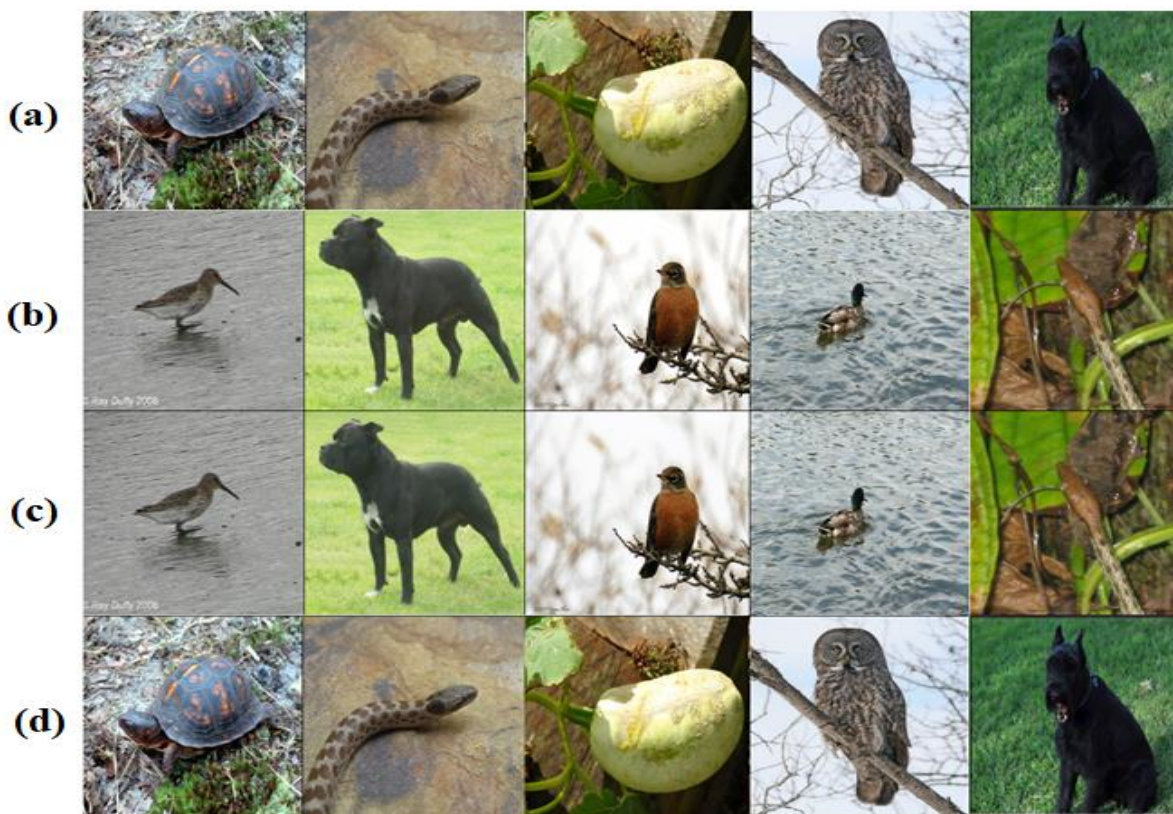
Figure 5. Steganographic Results for samples of images within video. (a) Input image, (b) Cover image (c) Container image "stego image" (d) Retretive image.

As shown in Fig. 5, the proposed system takes two images, one is the secret images (first row), and the second is the cover image (second row). The task is to hide the secret images inside the cover images (the second row). Then, the Hided network output is called the container image (the third row). After that, this container image is then can be passed to the RN, which is able to retrieve the hidden image, and this is known as a reverse secret image (the Fourth row). From the result, it is tough to notice the visual difference between the

cover and the container images. However, the reveal network can get back the data of the secret image only with a minimal deviation.

The Loss Curves

The loss curves of the hiding and revealing are computed based on equation 1, where the beta value is (0.75), and the batch size is 32 (represent 16 covers and 16 secrets). The results are shown in Fig.6.

(a) (b)
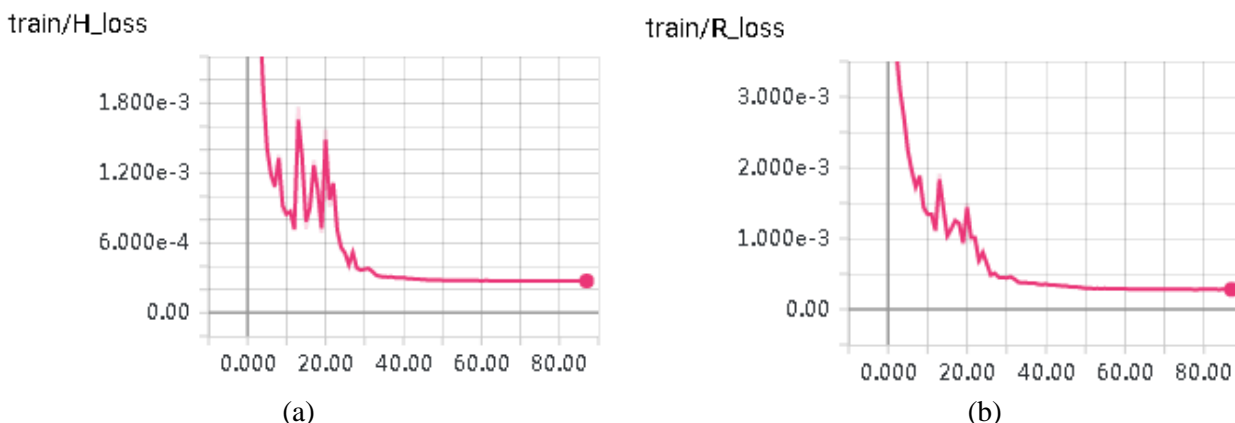
Figure 6. Loss curves for, (a) Hiding net loss (H_loss), (b) Revealing net loss (R_loss)

From Fig. 6, the two networks are trained, hiding net (H) and revealing net (R) with values, 0.75 beta, and 32 batch size. From the figure, the

mean squared error (MSE) loss with stochastic gradient descent against is drawn against the training epoch for (a) hiding net and (b) the

revealing net. The results show that the MSE is reduced through training epoch and reaches to lower value with stability in about (50) epoch, which reaches to value about (0.3. The validation error tracks the training error without upward divergence, demonstrating a stable training regime with a good bias-variance tradeoff.

The averaged pixel-wise discrepancy is described in Table 2.

Table 2. The averaged pixel-wise discrepancy.

Dataset	Container - Cover (APD) (0-255)	Secret - Rev_Secret (APD) (0-255)
Training	4.20	4.73
Validation	4.16	4.40

Similarity, Cryptanalysis, and Attacking

In this part, the cover is tested for similarity. Form the test results, and it is visually complicated to observe any difference between the cover and the container image. For attacking and cryptanalysis, even if the attacker had the ability to obtain several instances of container images that were created by the proposed system, it is difficult to recover the hidden image. This because the model is explicitly minimizing the similarity of the cover image is compering to the hidden image. This is related to the mechanisms of the proposed system. In which the pixels are permuted in one of M before hiding the secret image. Then the permuted-secret-image will then be hidden by the system, as is the key (which is an index to M). This leads to making a recovery hard even by searching at the residuals (If assuming that theattacker can access the original image) because the residuals do not have any spatial structure.

Conclusion:

In this work, a video steganography method has been designed based on utilized CNN, and the main goal is to achieve improvement in some stego aspects, including capacity, visibility, and higher resistance to cryptanalysis attack. From results it can point the following:

- •As the model involves two inputs and outputs, thus, a custom generator had being used to feed the system with input images (from video file) and cover image from the cover image directory. The output labels for the system is corresponding to 2 input images of the system, in every iteration.
- For loss function, the hidden loss and revealed loss should both be computed and then add them up by using customized loss weights. Additionally, these loss functions need to pass as customized objects for prediction through runtime.
- In case that both networks are trained together, they should be split to to hide and reveal networks. To achieve this, the initial block (encoder) is separated from the parent model via determination of the requested intermediate layer as it is the final output layer. Alternatively, a new input layer has been used to feed the decoder part, and then connected it with the lower layers (with weights). This has been achieved by re-initializing all these layers (with the same name) and then reloading the related weights from the parent model.
- As the proposed model support just the lossless formats, it need have to make sure that the output container image will be modified before decoding. In addition, there is a need to be sure not to save the container video by using a lossless (uncompressed) codec format.
- A permutation-based block shuffling approach has been used for encrypting the secret images. This has been achieved by dividing the image into fixed-size blocks and permute all of them depending on the predefined sequence that provides a shared secret key prior to hiding them by using the model. The receiver needs to use this secret key to be able to decode the secret image from reveal output.
- In order to make training faster and/or convergence, larger batch size should be used (better GPU) and the learning rate decay.
- As a regression task are mainly performing, thus, the accuracy has limited meaning regarding autoencoder networks. However, KL divergence or MSE (on hidden data) can be usesss as metrics to determine the performance.

Authors' declaration:

References:
1. Sensarma D, Sarma SS. Data Hiding using Graphical Code based Steganography Technique. arXiv preprint

arXiv:1509.08743. 2015 Sep 29.

2. Warf B, editor. The SAGE Encyclopedia of the Internet. Sage; 2018 May 15.

3. Yadav RM, Tomar DS, Baghel RK. A Study on Image Steganography Approaches in Digital Images EUSRM. 2014 May;6(5):1-6..

4. Shi YQ, Kim HJ, Pérez-González F, Yang CN, editors. Digital-Forensics and Watermarking: 13th International Workshop, IWDW 2014, Taipei, Taiwan, October 1-4, 2014. Revised Selected Papers. Springer; 2015 Jun 24.

5. Mishra M, Mishra P, Adhikary MC. Digital image data hiding techniques: A comparative study. arXiv preprint arXiv:1408.3564. 2014 Aug 15.

6. Hmood AK, Zaidan BB, Zaidan AA, Jalab HA. An overview on hiding information technique in images. JApSc. 2010 Dec;10(18):2094-100.

7. Meng R, Cui Q, Yuan C. A survey of image information hiding algorithms based on deep learning. CMES. 2018 Dec 1;117(3):425-54..

8. Weng X, Li Y, Chi L, Mu Y. High-Capacity Convolutional Video Steganography with Temporal Residual Modeling. InProceedings of the 2019 on International Conference on Multimedia Retrieval 2019 Jun 5 (pp.87-95).

9. Rawat W, Wang Z. Deep convolutional neural networks for image classification: A comprehensive review. Neural Comput. 2017 Sep;29(9):2352-449.

10. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018 May 1;77:354-77.

11. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Wang L, et al. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434. 2018 Dec 20.

12. Yedroudj M, Comby F, Chaumont M. Yedroudj-net: An efficient CNN for spatial steganalysis. In2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018 Apr 15 (pp. 2092-2096). IEEE.

13. Salomon M, Couturier R, Guyeux C, Couchot JF, Bahi JM. Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key: A deep learning approach for telemedicine. European Research in Telemedicine/La Recherche Européenne en Télémédecine. 2017 Jul 1;6(2):79-92.

14. Baluja S. Hiding images in plain sight: Deep steganography. InAdvances in Neural Information Processing Systems, 2017 (pp. 2069-2079)..

15. Zhu J, Kaplan R, Johnson J, Fei-Fei L. Hidden: Hiding data with deep networks. InProceedings of the European conference on computer vision (ECCV) 2018 (pp. 657-672).

16. Weng X, Li Y, Chi L, Mu Y. High-Capacity Convolutional Video Steganography with Temporal Residual Modeling. InProceedings of the 2019 on International Conference on Multimedia Retrieval 2019 Jun 5 (pp. 87-95).

17. Prasad K R. The Design and Development of Data Hiding Using Deep Learning. JASRAE. 2019;16(5): 970-974.

18. Kurak Jr CW, McHugh J. A cautionary note on image downgrading. InACSAC 1992 Dec 4 (pp. 153-159).

19. Hernandez-Castro JC, Blasco-Lopez I, Estevez-Tapiador JM, Ribagorda-Garnacho A. Steganography in games: A general methodology and its application to the game of Go. Comput Secur. 2006 Feb 1;25(1):64-71.

20. Hosmer C. Discovering hidden evidence. JDFP. 2006 Mar 1;1(1):47-56.

21. Khan I, Verma B, Chaudhari VK, Khan I. Neural network based steganography algorithm for still images. InINTERACT-2010 2010 Dec 3 (pp. 46-51). IEEE.

22. Qian Y, Dong J, Wang W, Tan T. Deep learning for steganalysis via convolutional neural networks. InMedia Watermarking, Security, and Forensics 2015 2015 Mar 4 (Vol. 9409, p. 94090J). International Society for Optics and Photonics.

23. Xu G. Deep convolutional neural network to detect J-UNIWARD. InProceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security. 2017 Jun 20 (pp. 67-73).

24. Tang W, Tan S, Li B, Huang J. Automatic steganographic distortion learning using a generative adversarial network. IEEE SPL. 2017 Aug 29;24(10):1547-51..

25. Zeng J, Tan S, Li B, Huang J. Large-scale JPEG image steganalysis using hybrid deep-learning framework. IEEE Transactions on Information Forensics and Security. 2017 Dec 4;13(5):1200-14..

26. Zhu J, Kaplan R, Johnson J, Fei-Fei L. Hidden: Hiding data with deep networks. InProceedings of the European conference on computer vision (ECCV) 2018 (pp. 657-672)..

27. Hussain I, He Q, Chen Z. Automatic fruit recognition based on dcnn for commercial source trace system. IJCSA. 2018;8.

28. Meng R, Cui Q, Yuan C. A survey of image information hiding algorithms based on deep learning. CMES. 2018 Dec 1;117(3):425-54.

29. Duan X, Guo D, Liu N, Li B, Gou M, Qin C. A New High Capacity Image Steganography Method Combined With Image Elliptic Curve Cryptography and Deep Neural Network. IEEE Access. 2020 Feb 4;8:25777-88.

30. Kishore DR, Suneetha D, Babu PN, Chinababu P. Deep Convolutional Neural Network-based Image Steganography Technique for Audio-Image Hiding Algorithm. IJEAT. 2020; 9(4):2187-2189.

إخفاء البيانات الذكية المتقدمة باستخدام الأخفاء بالفيديو والشبكات العصبية الألتفافية

أيمان سليم أبراهيم حربه[1]   هند سليم أبراهيم حربه[2]   ايناس علي عبدالمنعم[3]

[1]وحدة ابن سينا للتعليم الاكتروني، كلية الاداب، جامعة بغداد، بغداد، العراق
[2]قسم علم الجو، كلية العلوم، الجامعة المستنصرية، بغداد، العراق
[3]قسم علوم الحاسوب، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة:

إخفاء المعلومات هو تقنية لإخفاء البيانات السرية ضمن ملفات أخرى من نفس النوع أو في أنواع أخرى. وتعد تقنية إخفاء البيانات من التقنيات الضروريه في أمن المعلومات الرقمية. يهدف هذا العمل إلى تصميم طريقة إخفاء المعلومات في الاتصال الإلكتروني (ستيجانوجرافيا) يمكنها إخفاء رسالة داخل صور ملف الفيديو بشكل فعال. في هذا العمل ، نحاول اقتراح نموذج إخفاء المعلومات بالفيديو من خلال تدريب نموذج لإخفاء الفيديو (أو الصور) داخل فيديو آخر باستخدام الشبكات العصبية الألتفافية (CNN). في النموذج المقترح يتم تدريب CNN بشكل متزامن لتوليد عمليات الكشف والاختباء ، وهي مصممة للعمل بشكل مزدوج (أي يتم تدريب الشبكتين بنفس الوقت). يحتوي هذا النموذج على إستراتيجية جيدة لأنماط الصور ، والتي تساعد على اتخاذ قرارات لتحديد أي أجزاء من صورة الغلاف يجب أن تكون زائدة عن الحاجة ، والتي تسمح بأخفاء المزيد من وحدات البكسل هناك. يمكن تنفيذ CNN باستخدام مكتبة (keras)، جنبًا إلى جنب مع مكتبة (tensorflow). بالإضافة إلى ذلك ، تم استخدام صور ملونه (RGB)عشوائية من مجموعة بيانات " ImageNet" لتدريب النموذج المقترح (حوالي 45000 صورة بالحجم (256 × 256)). تم تدريب النموذج المقترح باستخدام صور عشوائية مأخوذة من قاعدة بيانات ImageNet ويمكنه العمل على الصور المأخوذة من مجموعة واسعة من المصادر. ومن خلال توفير مساحة على الصورة عن طريق إزالة المساحة الزائدة ، يمكن زيادة كمية البيانات المخفية (تحسين السعة). ونظرًا لأن الأوزان معمارية النموذج يتم توزيعهما بشكل عشوائي ، فلا يمكن معرفة الطريقة الفعلية التي ستخفي الشبكة بها البيانات لأي شخص ليس لديه الأوزان. علاوة على ذلك ، تم التشفير بطريقة خلط الكتلة (block-shuffling) كطريقة تشفير لتحسين الأمان ؛ وأيضا تم استخدام طرق تحسين الصورة لتحسين جودة الصور الناتجة. واظهرت النتائج ، أن الطريقة المقترحة حققت مستوى أمان عالي، وقدرة تضمين عالية. بالإضافة إلى ذلك ، اظهرت النتائج أن النظام حقق نتائج جيدة في إمكانية الرؤية والهجمات، حيث نجحت الطريقة المقترحة في خداع المراقب وبرنامج تحليل ستيجانوجرافيا.

الكلمات المفتاحية: الشبكات العصبية الألتفافية، أخفاء البيانات، الأخفاء بالصور، إخفاء المعلومات في الاتصال الإلكتروني، الأخفاء بالفيديو.