ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse75942020.pdf
https://doi.org/10.30534/ijatcse/2020/75942020

Image Steganography by Deep CNN Auto-Encoder Networks

Ismail Kich [1], El Bachir Ameur[1], Youssef Taouil[1], Amine Benhfid[1]
[1] Research Team MSISI – LaRIT, Department of Computer Science, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco
kichsma@gmail.com, ameurelbachir@yahoo.fr, taouilysf@gmail.com, aminebenhfid@gmail.com

ABSTRACT

The existing traditional image steganography methods often adopt the selection and mapping approaches. Among all the pixels of the cover image, only those which have the portability of incorporating the secret bits without noticeable distortion are chosen. This results to small integration capacity. In this paper, we propose a generic system of image steganography that uses the architecture of auto-encoding networks based on end to end trained deep Convolutional Neural Networks to ensure the process of concealment and extraction. The trained network includes two sub-networks, one for hiding used by the sender to encode a color image in another of the same size. The other for extraction, used by the recipient to retrieve the secret image from the received stego image. To validate our system, we carried out several tests on a range of challenging images dataset publicly available such as ImageNet, CIFAR10, LFW, PASCAL-VOC12. Results show that the proposed method is generic regardless the source of the images used and solves the problem of capacity at acceptable PSNR and SSIM values.

Key words: Information security, Image steganography, Deep Neural Network, CNN, Encoder-Decoder.

1. INTRODUCTION

Man has always sought to protect his important and sensitive data and to make his most important communications confidential to others. Nowadays, with the development of the internet and its applications such as cloud computing and enormous online storage capacities, individuals and organizations are allowed to process, access to and exchange data. Hence emerges the need to develop new mechanisms and technologies to protect data from theft and interception by unauthorized parties [1], [2].

Cryptography and Steganography are two common techniques which are used to remedy this problem. In Cryptography, encryption uses certain algorithms to make data incomprehensible to unauthorized people, but still this encryption attracts attention from eavesdropper which arises the possibility that secret data may be hacked. In steganography, the idea is to hide the secret message into innocent looking media carriers, this does not create any suspicion of third-party [3], [4]. This technique allows to transmit the data undetectably; and even more, make the secret unintelligible if it is detected. Multimedia files such as video, image, sound, text, protocol ..., are used as a cover object in steganography. However, the image remains the most used as cover object in academic research. Image steganography finds its application in many areas such as watermarks, confidential data transmission, copyright certification, integration of patient's data into their scanner images and many other applications [5], [6].

In general, imperceptibility, capacity and security against different attacks are the criteria with which one we can measure the performance of a steganographic model. Imperceptibility allows us to estimate the similarity between the cover image and its corresponding stego image. Capacity refers to the average number of bits inserted in each pixel of the image cover; it is measured by bit per pixel (*bpp*). Security expresses the possibility to identify the stego image from natural images by third-party steganalysis attacks. Consequently, favoring one parameter over the others influences the performance of a steganographic model. The more secret data the stego image hides, the more the quality of the stego image is degraded and may become noticeable [7]. The choice of the cover image also plays a very important role in the security of the stego image; an image rich in noisier and edge regions allows data concealment without detectable disturbance than an image rich in smooth regions. It is therefore essential to seek a compromise between the values of these parameters, and especially one which makes it possible to obtain good capacity while retaining acceptable values of the other parameters [8].

Because of their simplicity, the Least Significant Bit substitution (LSB) methods are extremely popular in image steganography. It hides directly the secret bits in the pixels of the cover image either in a uniform or adaptive manner, by simple substitution or by more improved versions of substitution. Although often the visual analysis cannot detect the distortion due to the stego image, the fact that the pixels of the cover image are modified disjointly causes a disturbance in the distribution of the LSB of the pixels of the stego image, thus making the image easily detectable by a statistical attack [9]–[11].

Alternatively, other researchers suggest searching the cover image for the pixels to be modified while preserving the statistical distribution of the image. Pixels in texture regions, edges, brightness are a better choice for hiding secret information than those in smooth regions. Work on estimating the payload that a cover image can provide while preserving robustness against statistical attacks can be found in [12], [13].

In [14], the authors use the S-UNIWARD or J-UNIWARD algorithm to incorporate more payload into the noisy or complex region of the cover image to obtain excellent invisibility and security of the stego image. The authors in [8], [15]–[18] use edge detectors such as Canny and fuzzy or hybrid detection to identify the pixels of the edge regions; data is hidden in these regions using improved LSB as a substitution method. Results have shown that this technique is able to increase the hidden payload while retaining a better quality of stego image with good robustness against statistical detection attacks. In [19], the authors propose a technique that combines the advantages of edge detection and XOR coding to increase the hidden payload and avoid detection during a steganalysis attack. In short, these works used different techniques to define the basic properties of the cover image and select the right places for the concealment of secret data while preserving its undetectability against visual attacks and its security against structural attacks.

Recently, and after the impressive results obtained by merging deep neural networks with steganalysis [20]–[23], researchers have attempted to incorporate deep neural networks to select LSBs of pixels where data is to be concealed. Others have used deep neural networks to determine the bits to be extracted from stego images to reveal secret message [24]–[26].

In steganographic models where the hidden secret message is a text, it is required at the recipient level that we extract the message perfectly (without error). This condition can be mitigated to some extent by using the image as hidden information. In [27], the author proposes an architecture inspired by image compression via auto-encoding networks [28] using CNN convolutional neural networks. The network of this architecture is composed of three subnets: Prep-Network, Hiding Network and Reveal Network. The first two subnets encode the secret image into the cover image so that the resulting stego image appears as similar as possible to the cover image. Both color secret and cover images have the same size. The third network allows to extract from the stego image the revealed image, it is very similar to the encoded secret image. Each network uses a sequence of 5 convolution layers that had 50 filters each of {3 x 3, 4 x 4, 5 x 5} patches. This model is different from other existing conventional steganography models. It provides large hiding capacity (24 *bpp*) with an acceptable invisibility. However, the architecture used in this method is considerably complicated and the color of generated stego images is distorted.

In [29], authors proposed a deep learning based generic encoder-decoder architecture to complete the same task, except that they used gray images as secret images. This architecture is composed only of two networks (encoder and decoder) which are end to end trained. The encoder network has two parallel branches: guest branch and host branch. The guest branch receives the secret gray image and uses a sequence of operations to decompose it into low-level and high-level features. The host branch uses a sequence of convolution operations to decompose the cover image into a hierarchy of features then fusion the extracted features of secret image into cover image. Each network uses a sequence of 3 x 3 convolution and ReLU layers, except the last layer which use 1 x 1 convolution without ReLU activation. Experimental results show that this method also offers a large hiding capacity (8 bpp) with good invisibility. But, the stego image still has the problem of color distortion.

The authors in [30] proposed an image steganography model based on the techniques of deep learning. In this model, a hiding network is employed to embed a color image into another color image of the same size; and an extraction network is used to extract the secret image concealed in the stego image produced by hiding network. The hiding network's architecture is similar to U-Net structured CNN which is composed of two phases: a contraction phase and an expansion phase. The contraction phase uses a sequence of 4 x 4 convolution layers followed by a Leaky ReLU activation function and Batch Normalization (BN) operation in each down-sampling. In the expansion phase, each up-sampling uses 4 x 4 deconvolution layers followed by a ReLU activation function and BN operation, except the output layer which use Sigmoid activation function to calculate the stego image. As for the extraction network, it uses a sequence of 3 x 3 convolutional layers, followed by a BN operation and a ReLU activation function, except the output layer which use the Sigmoid activation function to calculate the revealed image. On the one hand, this method has significant advantages in terms of capacity (24 *bpp*) and invisibility. On the other hand, the architecture of this method is not generic.

In summary, although these works provided a good compromise of invisibility and hiding capacity, they still have problems of color distortion or they are not generic. In our contribution, we tried different networks structures and successfully avoided these problems while reaching a better compromise of invisibility and capacity. we propose in this paper a generic encoder-decoder architecture based on deep learning for image steganography. The goal is to hide a color image of N x N x RGB in a color image of the same size without causing significant and visible distortion to it. The

deep neural network decides where the secret bits will be embedded and dispersed in the cover image and how they are actually encoded. Our approach is strongly inspired by image compression systems via auto-encoding networks [28], where the system must learn to compress the secret image in different parts of the cover image using networks of neurons and guaranteeing a better quality for the stego image and the extracted hidden image.

The rest of this article is organized as follows: Section 2 gives details on the architecture of the convolutional neural networks (CNN) of the auto-encoding network and describes the proposed method. Section 3 presents the experimental results and analysis. The conclusions are presented in section 4.

2. METHODOLOGY

Instead of using the classical data concealment algorithms such as LSB substitution algorithm and its improved versions, the authors in [27] proposed image steganographic model that uses deep learning based on generic encoder-decoder architecture. The location where to hide data is selected by ingenious networks of neurons; the network structure of this deep model is composed of three sub-networks: Pre-network, Hiding network and Reveal network. The structure of the network is shown in Figure 1. The pre-network has as input the hidden image, its role is to preprocess the secret image. firstly, if the size of the secret image is less than that of the cover image, then the preprocessing will distribute the bits of the original M x M image at N x N (size of the cover image). Secondly, it is important to transform the color-based pixel functions into more useful features like the Edge information to facilitate the encoding of the secret image. The Hiding Network is the main masking network, it takes as input the cover image and the pre-network's output. As result a container image called the stego image. The input size of this network is N x N x RGB (original size of the cover image) plus the transformed channels of the secret image. Finally, the third network is Reveal network through which the receiver can decode the secret image from the stego image.
Unlike [27], our model uses only two networks: the hiding network and the reveal network. We train end to end the two networks using convolutional neural networks to create a container image (Stego image) from a pair of input images of the same size (Cover and Secret images) and extract the hidden image from the Stego image. Figure 2 shows the proposed architecture in detail.

CNN layers are used to learn the hierarchy of image features, a hierarchy ranging from low-level generic features to high-level specific features. Thus, the encoder network learns the features of the two images which allows it to hide the details of the image to be hidden in the features of the cover image. In other words, the objective is to compress and distribute the bits of secret image on all the bits available on the cover image. In our method, the hiding network trains itself to hide a secret image in a cover image of the same size in order to produce an output container image with little distortion as possible so that it remains visually identical to the cover image. At the same time, the reveal network is training to extract the secret image from the stego image produced by the hiding network.

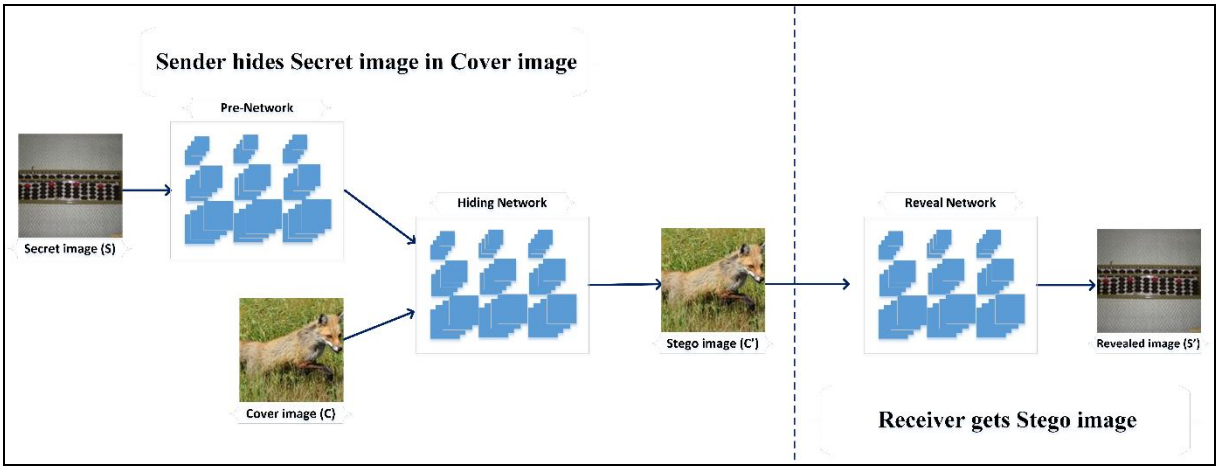


Figure 1: Image steganography architecture based on DNN

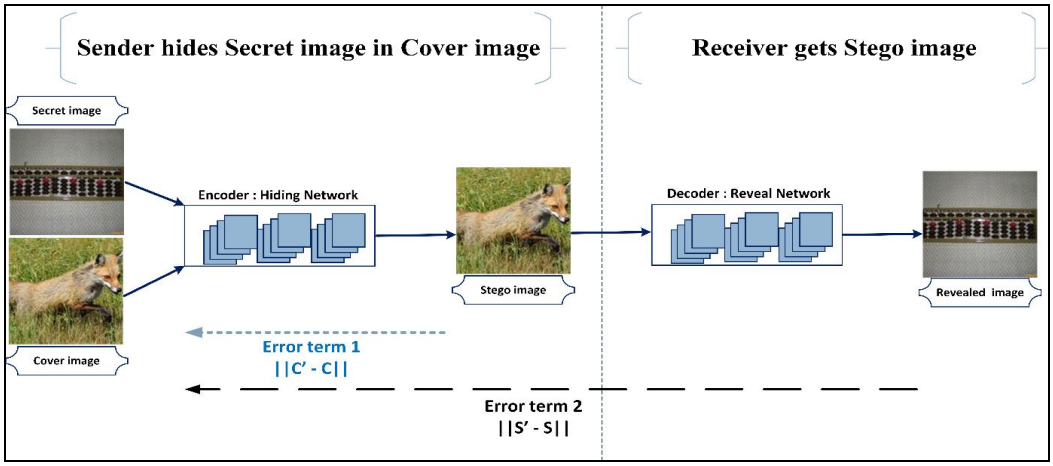


Figure 2: Architecture of proposed method

2.1 Hiding network Architecture

As illustrated in Figure 3, the encoder network of our method is designed as plain network receiving as input the cover image and the secret image, both are concatenated into a 6-channel tensor. it is made up of two phases. The first phase of the network is designed with a sequence of 3 x 3 convolution layers, each convolution is followed by a BN operation to accelerate learning and a ReLU activation function. We start with 64 feature channels and we double the number of feature channels after each convolution. After four convolution operations, the number of feature channels is 512. In the 2nd phase, the feature map is oversampled using also a sequence of 3 x 3 convolution layers followed by a BN operation and a ReLU activation function. At the same time, each oversampling operation is cascaded with the characteristics map of the corresponding stage of the first phase so that the network in this phase learns the functional maps of the previous stages. At the last network layer, the 3 x 3 convolution is applied to compress the convolved feature channels into a 3-channel characteristic map followed by a BN operation and a Sigmoid activation to calculate the output which in this case is our container image or Stego image.

2.2 Reveal network Architecture

As illustrated in the Figure 4, the reveal network is also designed as plain network, it receives as input the stego image produced by the encoding network. we apply to this image a sequence of layers of 3 x 3 convolution; each convolution is followed by a BN operation to accelerate the training and a ReLU activation function. But, in the last layer, a Sigmoid activation is applied to compress the convoluted features channels into a 3-channel features to calculate the secret image (output).

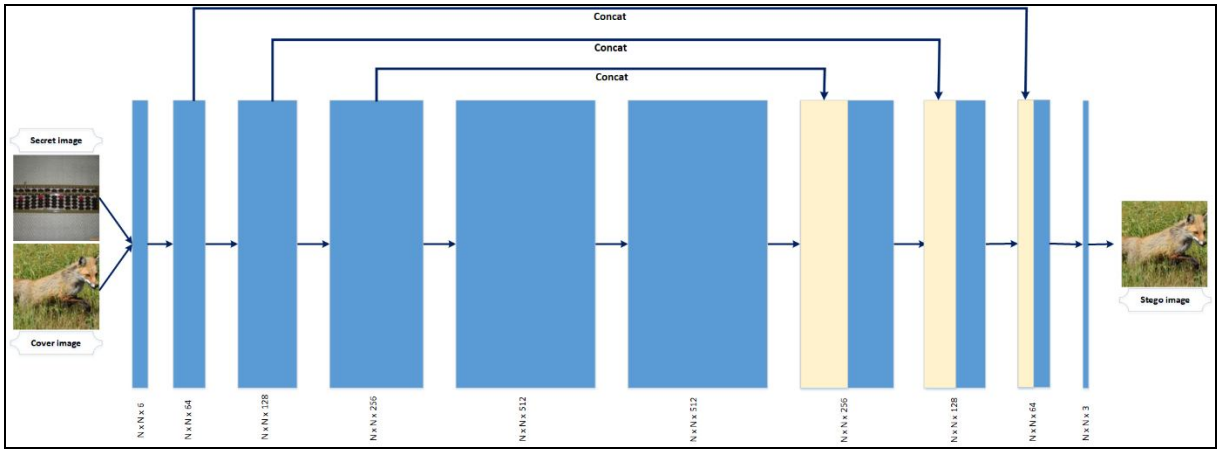


Figure 3: Hiding network scheme

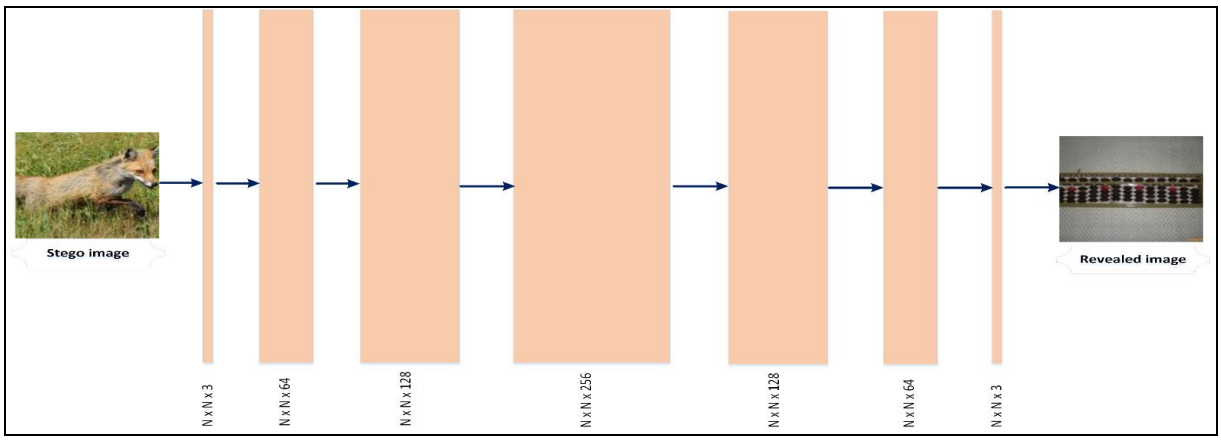


Figure 4: Reveal network scheme

2.3 Loss function

Like in [27], the proposed auto-encoding networks is used to encode two images (cover and secret) so that the container image (stego image) produced appears as similar as possible to the cover image, and that the revealed image produced by the reveal network also appears similar to the original secret image at a certain threshold β defined by the user.

Let C and S represent the cover and secret input images for the Hiding network respectively, while C' and S' represent the Stego image and the revealed image respectively. The two networks are trained end to end using the loss function as follows:

$$L(C,C',S,S') = \|C - C'\| + \beta \|S - S'\| \qquad (1)$$

$\|C\text{-}C'\|$ and $\|S\text{-}S'\|$ are respectively the costs of the hiding network and the reveal network. The weights of the error term of the Hiding network is not shared with the weights of the reveal network, whereas the weights reveal network are shared across the entire auto-encoding networks. β is how to weigh their reconstruction errors. Training the two networks using this equation ensures that the features of the secret image are fully encoded on the cover image. Both networks were trained using Adam as an optimization method [31], to minimize the Sum of Squares Error of the pixel difference in the reconstructions.

3. EXPERIMENT RESULT

In this part, we'll present and discuss the results of our experiments. Image Datasets like ImageNet [32], CIFAR10 [33], LFW [34] and PASCAL-VOC12 [35] have been set up to test our training network system. Each database is randomly divided into three datasets, namely: training, validation and test. All training results have been validated by the validation set and the results reported in this document are performed on the test set. The Adam learning method is used with an initial learning rate set to 0.001 and was descended to 0.0001 after 30 epochs of training. All weights of our model were initialized randomly using the Xavier initialization [36]. The number of images per batch is fixed at 16. The value β is fixed at 0.75, since in our system we do not need to completely reveal the secret image. Of course, the parameter β can be set to a higher value to have a better quality of the revealed secret image.

We use the Peak Signal-to-Noise Ratio (PSNR) [37] and the Structural Similarity Index (*SSIM*) [38] as metrics to measure our proposed model's performance. The *PSNR* checks out image imperceptibility by calculating the error between corresponding pixels. The more the cover and stego image are close, the higher is the value of *PSNR*. It is calculated as follows:

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right)$$

$$MSE = \frac{1}{CL} \sum_{i=1}^{C} \sum_{j=1}^{L} \left(p_{i,j} - p'_{i,j} \right)^2$$

Where $p_{i,j}$ and $p'_{i,j}$ are the pixels intensity of cover and Stego images, *C* and *L* are the width and height of the cover image.

SSIM estimates the similarity of the image by calculating three terms namely: luminance, contrast and structure. The closer the cover image and the stego image, the closer the value of *SSIM* is to 1. The *SSIM* is calculated using the following formula:

$$SSIM(x,y) = l(x,y)c(x,y)s(x,y)$$

Where

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},\ c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},\ s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

μ_x is the average of *x*; μ_y is the average of *y*; σ_x^2 is the variance of *x*; σ_y^2 is the variance of *y*; $\sigma_{x,y}$ is the covariance of *x*, *y*; C_1, C_2 and C_3 are the constants to stabilize the division weak denominator.

For the first experiment, pairs of secret-cover images are randomly sampled from the CIFAR10 dataset which contains 50,000 size color images (32 x 32 x 3). We were able to hide

one image in another, so hide a payload of 100% (24 *bpp*) in the cover image with acceptable values of *PSNR* and *SSIM*. As shown in table 1, on the training set, the average values of *PSNR* and *SSIM* for images generated by hiding network are respectively 36.85 *dB* and 0.9855; as for the images produced by the reveal network, the values are 33.66 *dB* and 0.9714.

For the second experiment, we used CIFAR10 dataset, as a source of cover images and ImageNet dataset as a source of secret images. Images from ImageNet were randomly selected and resized to 32 x 32 x 3 (same size in CIFAR10). On the training set, the average values of *PSNR* and *SSIM* are respectively 36.01 *dB* and 0.9824 for Stego images, and 32.75 *dB* and 0.9683 for the revealed images. Almost the same results are obtained by reversing the sources of cover and secret images, as shown in Table 1. In all tables, The *PSNR* in the cover-stego column measures the invisibility of the model, whilst the *PSNR* in the Hidden-extracted columns measures the similarity between the hidden and revealed secret image, it evaluates the integrity of the model.

Table 1: *PSNR* and *SSIM* values for different runs of the proposed algorithm on different datasets.

Exp.	Cover Dataset	Secret Dataset	Cover-stego		Secret-revealed	
			PSNR	SSIM	PSNR	SSIM
1	CIFAR10	CIFAR10	36.85	0.9855	33.66	0.9714
2	CIFAR10	ImageNet	36.01	0.9824	32.75	0.9683
3	ImageNet	CIFAR10	35.55	0.9816	32.74	0.9673

From these experiences, we can say that the proposed auto-encoding network is extremely generic, and that we can use the same architecture to reliably guarantee the concealment of a secret image in another cover image of the same size with acceptable *PSNR* and *SSIM* values.

To test the performance of our system on large images, we designed two other experiments on the ImageNet dataset. The first experiment concerns images of size 128 x 128, therefore 30,000 images were randomly selected to form pairs of secret-cover images of the training, validation and test sets. All these images were then resized to 128 x 128 x 3. The average values of *PSNR* and *SSIM* respectively are 36.00 *dB* and 0.9692 for container images, and 33.33 *dB* and 0.9445 for the revealed secret images as can be seen in Table 2. In the second experiment, limited by the computing power, 10,000 images were selected randomly and resized to 256 x 256 x 3 to form pairs of secret-cover images. The batch size was set at 4. The average values of *PSNR* and *SSIM* respectively are 35.22 *dB* and 0.9554 for stego images, and 32.21 *dB* and 0.9338 for the revealed secret images.

Table 2: *PSNR* and *SSIM* values for large images from ImageNet dataset.

Exp.	Size	Cover-stego		Secret-revealed	
		PSNR	SSIM	PSNR	SSIM
1	128 x 128 x 3	36.00	0.9692	33.33	0.9445
2	256 x 256 x 3	35.22	0.9554	32.21	0.9338

To further verify the portability of our system on other images from different sources, we have run our algorithm trained by ImageNet on samples of images from PASCAL-VOC12 [14] and from Labelled Faces in Wild (LFW) [13]. 5000 images were randomly selected from each dataset to form pairs of secret-cover images of test sets. From the results of this experiment shown in Table 3, we can say that our algorithm can hide images in other cover images with good values of *PSNR* and *SSIM* regardless the source of these images.

Table 3: *PSNR* and *SSIM* values of our ImageNet trained algorithm on FLW and VOC2012 datasets.

Images size	Cover Dataset	Secret Dataset	Cover-stego		Secret-revealed	
			PSNR	SSIM	PSNR	SSIM
128 x 128 x 3	LFW	LFW	38.47	0.9686	35.19	0.9484
	P.-VOC12	P.-VOC12	36.18	0.9709	33.68	0.9472
	LFW	P.-VOC12	38.37	0.9678	33.73	0.9467
256 x 256 x 3	LFW	LFW	38.16	0.9491	34.41	0.9419
	P.-VOC12	P.-VOC12	35.23	0.9588	32.38	0.9349
	LFW	P.-VOC12	38.37	0.9513	32.56	0.9384

For a qualitative check of our model. The figure 5 illustrates a sample of images from LFW and VOC2012 datasets as well as their corresponding produced images. From this figure, we can see that the proposed model, even if it is trained on ImageNet dataset, it is able to hide and recover images from different sources while preserving their imperceptibility.

(a) (b) (c) (d)

Figure 5: Sample results of the proposed algorithm on LFW (top two rows), PASCAL-VOC12 (last two rows) datasets. (a) cover images, (b) Secret images, (c) Stego images, (d) Revealed images.

To make sure that our auto-encoding network doesn't just encode the bits of the secret image in the LSBs of the cover image. We examined changes to the four pairs of cover-secret images randomly sampled from an ImageNet dataset during

different training phases. Figure 6 illustrates the results of steganography on images produced during the first training iterations. at the start of training, the value of the loss function is very important, and we note that the hidden secret image features are clearly perceptible by the human eye on the reconstructed images. Figure 7 shows the results during the stabilization phase, the loss value is minimized; the reconstructed images are almost identical, and it is difficult to perceive modifications in comparison to the origins. To observe the effect of the proposed steganographic model on the pixel distribution of the images, two other cover images are randomly sampled from ImageNet. Figure 8 illustrates the histograms of these images before and after the steganography process. The histograms show that the distribution of the pixel values of the produced images is practically the same as original images. we can therefore say that the auto-encoding network is trained to encode the information of the secret image in each pixel of the cover image instead of simply modifying the LSBs of the pixels. Hence, the proposed steganographic system is robust to visual and statistical attacks.

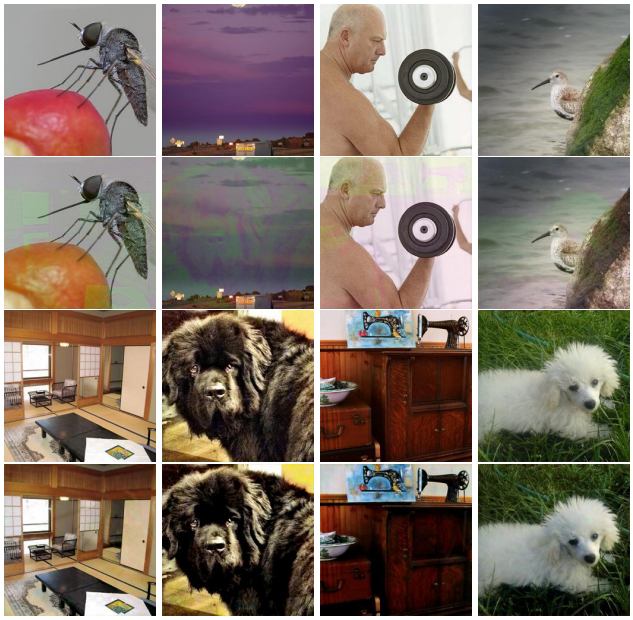


Figure 6: Sample results of the proposed algorithm during first iterations. First row represents Cover images, second the Stego images, third the secret images and fourth row represents the revealed images.

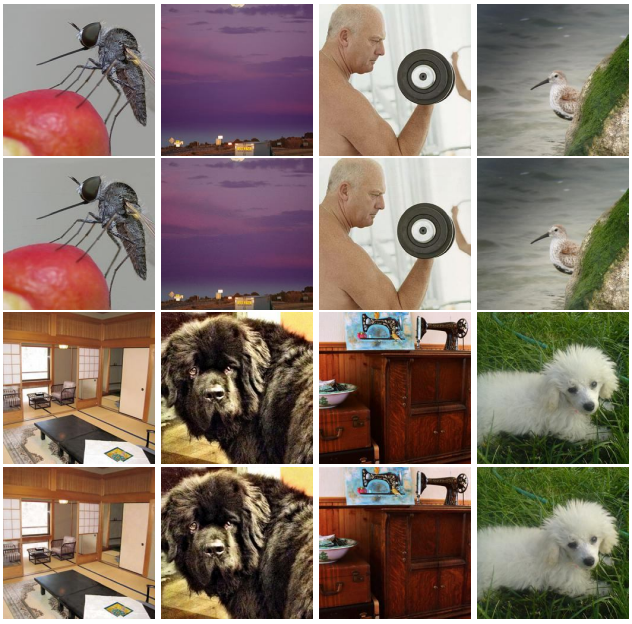
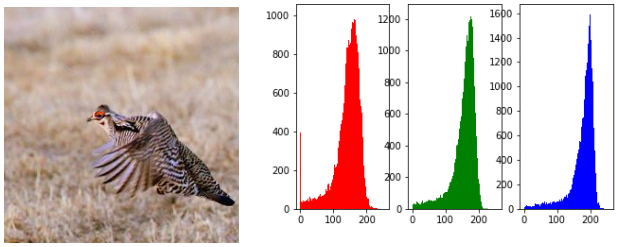
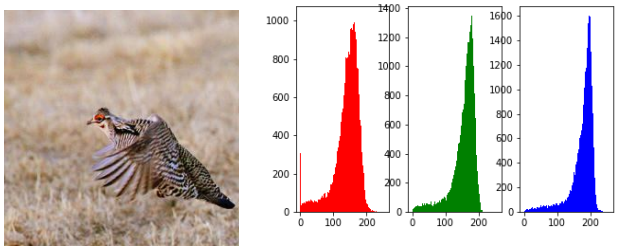


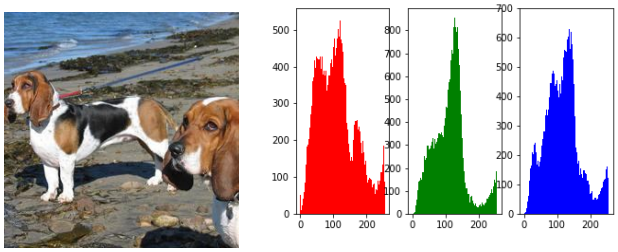
Figure 7: Sample results of the proposed algorithm during stabilization phase. First row represents Cover images, second the Stego images, third the secret images and fourth row represents the revealed images.



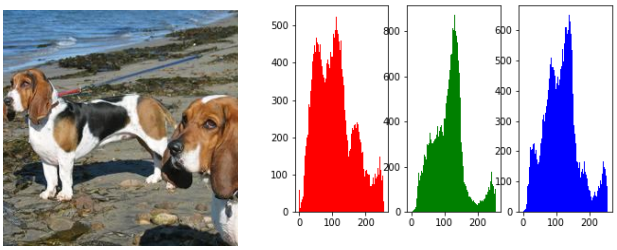
(a)



(b)



(c)



(d)

Figure 8: covers images before and after steganography. (a) cover image and its histograms, (b) Stego image and its histograms, (c) cover image and its histograms, (d) stego image and its histograms.

The following section describes the results of comparison of our method to those of the methods cited in [29] and [39] on different Datasets. We note that the Rehman's model and Zhang's model use gray images as secret images, ie a masking capacity of 33% (8 *bpp*). Table 4 represents the results of our model and those of the Rehman model applied to tiny images. Results show that the proposed model can encode a color image into another one of the same sizes, thus providing a hiding capacity of 100% (24 *bpp*). At the same time, the invisibility criterias of the proposed work are better; the PSNR is greater by 2 *dB* for ImageNet and 6 *dB* for CIFAR10. The SSIM of the proposed work is very closer to its optimal value 1 than the Rehman's method. Table 5 compares our model on large images to Rehman's and Zhang's models. These results were obtained by running the ImageNet trained models on sample of 1000 images from LFW and PASCAL-VOC12 datasets. Results show once more that our method guarantees a larger concealment payload of data while keeping a good imperceptibility. which is a sign of a reassuring invisibility since the stego image is subject to the possibility of monitoring and control by steganalysts. Hence, the proposed work outperforms Rehman's method and Zhang's model in both capacity and invisibility.

Table 4: Comparison of *PSNR*, *SSIM* and capacity values of the proposed method with the Rehman's model on 32 * 32 size images from different datasets.

Model	Cover Dataset	Secret Dataset	Cover-stego		Secret-revealed		Capacity %
			PSNR	SSIM	PSNR	SSIM	
Rehman's model	CIFAR10	CIFAR10	30.9	0.98	29.9	0.96	33
Proposed model	CIFAR10	CIFAR10	**36.85**	**0.9855**	**33.66**	**0.9714**	**100**
Rehman's model	ImageNet	ImageNet	32.9	0.96	36.6	0.96	33
Proposed model	ImageNet	ImageNet	**34.88**	**0.9825**	32.45	**0.9678**	**100**

Table 5: Comparison of *PSNR*, *SSIM* and capacity values of the proposed method with the Rehman's model and Zhang's model on 300 * 300 size images from different datasets.

Model	Cover Dataset	Secret Dataset	Cover-stego		Secret-revealed		Capacity %
			PSNR	SSIM	PSNR	SSIM	
Rehman's model	LFW	LFW	33.7	0.95	**39.9**	0.96	33
Zhang's model	LFW	LFW	34.63	0.9573	33.63	0.9429	33
Proposed model	LFW	LFW	**37.52**	**0.9597**	33.75	0.9498	**100**
Rehman's model	PASCAL-VOC12	PASCAL-VOC12	33.7	0.96	**35.9**	0.95	33
Zhang's model	PASCAL-VOC12	PASCAL-VOC12	34.49	0.9661	33.31	0.9467	33
Proposed model	PASCAL-VOC12	PASCAL-VOC12	**34.89**	**0.9667**	33.43	0.9465	**100**
Rehman's model	PASCAL-VOC12	LFW	33.8	0.96	**37.7**	0.95	33
Zhang's model	PASCAL-VOC12	LFW	34.45	0.9647	37.59	0.9495	33
Proposed model	PASCAL-VOC12	LFW	**34.91**	**0.9673**	34.79	0.9493	**100**

4. CONCLUSION

In this paper, we proposed a new steganographic system to hide one color image into another of the same size with minimal distortion of the images produced. The system uses the auto-encoding networks architecture based on end-to-end trained deep CNN to ensure the process of concealment and extraction. The experiment results prove that our method can be considered as a generic method by which various sources of images can be used, while guaranteeing an acceptable quality in terms of imperceptibility and similarity of the images produced compared to the originals. Limited by the computing power, we found difficulties in testing our method on large images (size more than 300 x 300 x 3). In the next step of this paper, we will try to exploit the possibilities that neural networks offer us to solve the problem of large images.

REFERENCES

[1] A. Nissar and A. H. Mir, "Classification of steganalysis techniques: A study," *Digital Signal Processing*, vol. 20, no. 6, pp. 1758–1770, Dec. 2010, doi: 10.1016/j.dsp.2010.02.003.

[2] Y. Taouil, E. B. Ameur, and M. T. Belghiti, "New Image Steganography Method Based on Haar Discrete Wavelet Transform," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, vol. 520, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International Publishing, 2017, pp. 287–297. https://doi.org/10.1007/978-3-319-46568-5_30

[3] B. Li, J. He, J. Huang, and Y. Q. Shi, "A Survey on Image Steganography and Steganalysis," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142–172, 2011.

[4] J. C. T. Arroyo, "An Improved Image Steganography through Least Significant Bit Embedding Technique with Data Encryption and Compression Using Polybius Cipher and Huffman Coding Algorithm," *IJATCSE*, vol. 9, no. 3, pp. 3376–3383, Jun. 2020, doi: 10.30534/ijatcse/2020/137932020.

[5] A. K. Hmood, B. B. Zaidan, A. A. Zaidan, and H. A. Jalab, "An Overview on Hiding Information Technique in Images," *J. of Applied Sciences*, vol. 10, no. 18, pp. 2094–2100, Dec. 2010, doi: 10.3923/jas.2010.2094.2100.

[6] M. Bachrach and F. Y. Shih, "Survey of Image Steganography and Steganalysis," in *Multimedia Security*, 1st ed., F. Y. Shih, Ed. CRC Press, 2017, pp. 201–214. https://doi.org/10.1201/b12697-11

[7] M. S. Subhedar and V. H. Mankar, "Current status and key issues in image steganography: A survey," *Computer Science Review*, vol. 13–14, pp. 95–113, Nov. 2014, doi: 10.1016/j.cosrev.2014.09.001.

[8] S. Kumar, A. Singh, and M. Kumar, "Information hiding with adaptive steganography based on novel fuzzy edge identification," *Defence Technology*, vol. 15, no. 2, pp. 162–169, Apr. 2019, doi: 10.1016/j.dt.2018.08.003.

[9] A. Benhfid, E. B. Ameur, and Y. Taouil, "Reversible steganographic method based on interpolation by bivariate linear box-spline on the three directional mesh," *Journal of King Saud University - Computer and Information Sciences*, Sep. 2018, doi: 10.1016/j.jksuci.2018.09.016.

[10] J. Fridrich, M. Goljan, and Rui Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE Multimedia*, vol. 8, no. 4, pp. 22–28, Dec. 2001, doi: 10.1109/93.959097.

[11] Al-Balqa' Applied University, Amman, Jordan and R. J. Rasras, "Comparative Analysis of LSB, LSB2, PVD Methods of Data Steganography," *IJATCSE*, vol. 8, no. 3, pp. 748–754, Jun. 2019, doi: 10.30534/ijatcse/2019/64832019.

[12] F. Yaghmaee and M. Jamzad, "Estimating Watermarking Capacity in Gray Scale Images Based on Image Complexity," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, p. 851920, doi: 10.1155/2010/851920.

[13] H. Yang, X. Sun, and G. Sun, "A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution," vol. 18, no. 4, p. 9, 2009.

[14] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. on Info. Security*, vol. 2014, no. 1, p. 1, Dec. 2014, doi: 10.1186/1687-417X-2014-1.

[15] I. Kich, E. B. Ameur, and A. Souhar, "New Image Steganography Method Based on K-means Clustering," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, Tetouan Morocco, Mar. 2017, pp. 1–6, doi: 10.1145/3090354.3090432.

[16] S. Islam, M. R. Modi, and P. Gupta, "Edge-based image steganography," *EURASIP J. on Info. Security*, vol. 2014, no. 1, p. 8, Dec. 2014, doi: 10.1186/1687-417X-2014-8.

[17] W.-J. Chen, C.-C. Chang, and T. H. N. Le, "High payload steganography mechanism using hybrid edge detector," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3292–3301, Apr. 2010, doi: 10.1016/j.eswa.2009.09.050.

[18] I. Kich, E. B. Ameur, and Y. Taouil, "Image Steganography Based on Edge Detection Algorithm," in *2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Kenitra, Dec. 2018, pp. 1–4, doi: 10.1109/ICECOCS.2018.8610603.

[19] H. Al-Dmour and A. Al-Ani, "A steganography embedding method based on edge identification and XOR coding," *Expert Systems with Applications*, vol. 46, pp. 293–306, Mar. 2016, doi: 10.1016/j.eswa.2015.10.024.

[20] Y. Qian, J. Dong, W. Wang, and T. Tan, "Deep learning for steganalysis via convolutional neural networks," in *Media Watermarking, Security, and Forensics*, 2015, vol. 9409, p. 94090J.

[21] J. Ye, J. Ni, and Y. Yi, "Deep Learning Hierarchical Representations for Image Steganalysis," *IEEE Trans.Inform.Forensic Secur.*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017, doi: 10.1109/TIFS.2017.2710946.

[22] S. Wu, S. Zhong, and Y. Liu, "Deep residual learning for image steganalysis," *Multimed Tools Appl*, vol. 77, no. 9, pp. 10437–10453, May 2018, doi: 10.1007/s11042-017-4440-4.

[23] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch," *Electronic Imaging*, vol. 2016, no. 8, pp. 1–11, 2016.

[24] S. Husien and H. Badi, "Artificial neural network for steganography," *Neural Comput & Applic*, vol. 26, no. 1, pp. 111–116, Jan. 2015, doi: 10.1007/s00521-014-1702-1.

[25] A. S. Brandao and D. C. Jorge, "Artificial Neural Networks Applied to Image Steganography," *IEEE Latin Am. Trans.*, vol. 14, no. 3, pp. 1361–1366, Mar. 2016, doi: 10.1109/TLA.2016.7459621.

[26] R. Jarušek, E. Volna, and M. Kotyrba, "Neural network approach to image steganography techniques," in *International Conference on Soft Computing-MENDEL*, 2016, pp. 317–327.

[27] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Advances in Neural Information Processing Systems*, 2017, pp. 2069–2079.

[28] G. E. Hinton, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[29] A. ur Rehman, R. Rahim, S. Nadeem, and S. ul Hussain, "End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography," in *Computer Vision – ECCV 2018 Workshops*, vol. 11132, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 723–729.

[30] 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, doi: 10.1109/ACCESS.2019.2891247.

[31] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[33] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[34] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," presented at the Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.

[35] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int J Comput Vis*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[37] A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, Aug. 2010, pp. 2366–2369, doi: 10.1109/ICPR.2010.579.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. on Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[39] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimed Tools Appl*, vol. 78, no. 7, pp. 8559–8575, Apr. 2019, doi: 10.1007/s11042-018-6951-z.