# End-to-End Trained CNN Encoder-Decoder Networks for Image Steganography

Atique ur Rehman(✉), Rafia Rahim(✉), Shahroz Nadeem(✉), and Sibt ul Hussain(✉)

Reveal (Recognition, Vision and Learning) Lab,
National University of Computer and Emerging Sciences (NUCES-FAST),
Islamabad, Pakistan
{atique.rehman,rafia.rahim,shahroz.nadeem,sibtul.hussain}@nu.edu.pk

**Abstract.** All the existing image steganography methods use manually crafted features to hide binary payloads into cover images. This leads to small payload capacity and image distortion. Here we propose a convolutional neural network based encoder-decoder architecture for embedding of images as payload. To this end, we make following three major contributions: (i) we propose a deep learning based generic encoder-decoder architecture for image steganography; (ii) we introduce a new loss function that ensures joint end-to-end training of encoder-decoder networks; (iii) we perform extensive empirical evaluation of proposed architecture on a range of challenging publicly available datasets (MNIST, CIFAR10, PASCAL-VOC12, ImageNet, LFW) and report *state-of-the-art* payload capacity at high PSNR and SSIM values.

**Keywords:** Steganography · CNN · Encoder-decoder
Deep neural networks

## 1 Introduction

In the field of information security steganography and steganalysis are considered as two important techniques [6,10]. Steganography is used to conceal secret information (*i.e.* a message, a picture or a sound) also known as payload into another non-secret object (that can be an image, a sound or a text message) also known as cover object, such that both the secret message as well as its content remain invisible.

In image steganography, most of the work has been done to hide a specific text message into a cover image. Thus the focus of all the existing techniques has been finding either noisy regions or low-level image features such as edges [7], textures [4], *etc.*, in cover image for embedding maximum amount of secret information without distorting the original image.


L. Leal-Taixé and S. Roth (Eds.): ECCV 2018 Workshops, LNCS 11132, pp. 723–729, 2019.
https://doi.org/10.1007/978-3-030-11018-5_64

In this work, we propose a novel and completely automatic steganography method for hiding one image to another. For this, we design a deep learning network that automatically identifies the best features from both cover and payload images to merge information. The biggest advantage of our this approach is that its generic and can be used with any type of images, to validate this we test our approach on variety of publicly available datasets including ImageNet, MNIST, CIFAR10, LFW and PASCAL-VOC12.

**Overall our main contributions** are as follows: (i) we propose a deep learning based generic encoder-decoder architecture for image steganography; (ii) we design a new loss function that ensures joint end-to-end training of encoder-decoder networks; (iii) we perform extensive empirical evaluation of proposed architecture on range of challenging publicly available datasets and report *state-of-the-art* payload capacity at high PSNR and SSIM values. Specifically, using our proposed algorithm we can reliably embed a unary channel image ($m \times n$ pixels) into a color image ($m \times n \times 3$ pixels). Our experiments show that we can achieve this payload of 33% (on average 8 bpp) with the average PSNR values of 32.9 db (SSIM = 0.96) for cover and 36.6 db (SSIM = 0.96) for recovered payload image.

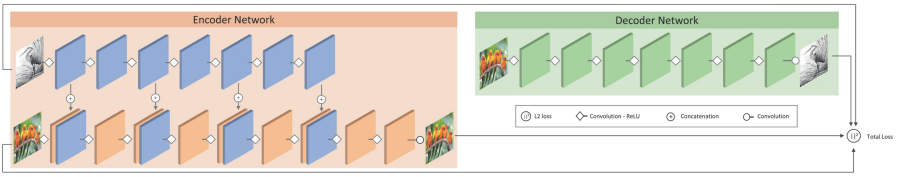


**Fig. 1.** Pictorial representation of encoder and decoder networks architecture. Top row in encoder network represents the guest branch while bottom row represents host branch.

## 2 Methodology

We train end-to-end a pair of encoder and decoder Convolutional Neural Networks (CNNs) for creating the hybrid image from pair of input images, and recovering the payload image from input hybrid image – *c.f.* Fig. 1 for architecture details. Here, we make use of observation that CNN layers learns a hierarchy of image features from low-level generic to high-level domain specific features. Thus our encoder identifies specific features from cover image to hide the details from the payload images, while decoder learns to separate those hidden features from the "hybrid" image.

Specifically, the encoder network takes two images (*i.e.* a "host" cover image and a "guest" payload image) as input and produces a single hybrid output image. Thus, the goal of encoder network is to produce a hybrid image, that remains visually identical to the host image but should also contain the guest

image content in it. The decoder network takes as input the encoder produced hybrid image and recovers the guest image from it. The goal of decoder network is to recover the guest image from the input hybrid that remains visually similar to input guest image of encoder.

Let $I_h$ and $I_g$ represent input host and guest images to encoder, while $O_e$ and $O_d$ represent the output hybrid image and output decoder image respectively, then the complete loss function for encoder and decoder network can be written as:

$$L(I_g, I_h) = \alpha||I_h - O_e||^2 + \beta||I_g - O_d||^2 + \lambda(||W_e||^2 + ||W_d||^2) \quad (1)$$

Here $W_e$ and $W_d$ represent the learned weights for the encoder and decoder networks respectively while $\alpha$ and $\beta$ are controlling parameters for encoder and decoder. The first term in loss function defines encoder loss and the second one decoder loss.

### 2.1 Encoder Architecture

The encoder network at the input end has two parallel branches named as guest branch and host branch. Guest branch receives the input guest image $I_g$ and uses a sequence of convolution and ReLU layers to decompose the input image into low-level (edges, colors, textures, *etc.*) and high level features. Host branch receives the input host image $I_h$ and uses a sequence of convolution and ReLU layers (except the last layer which does not include ReLU layer) to decompose the input image into a hierarchy of feature representations and merge the extracted representations of guest image into host image.

Precisely, for merging the information from guest image, encoder concatenates the extracted feature maps from each alternating layer of guest branch (starting from input) to the corresponding output features maps of host branch. This procedure is repeated up to a layer of depth $k$ (we found $k = 7$ as the best parameter), at this point we completely merge the guest branch features into host branch and guest branch cease to exist. After merging a further sequence of convolution and ReLU layers are used before the final convolution layer which produces as output hybrid image $O_e$.

### 2.2 Decoder Architecture

Our decoder network receives the encoder produced hybrid image $O_e$ as input and runs it through sequence of convolution and ReLU layers (except the last layer which does not include ReLU) to recover the concealed representation $O_d$ of guest image $I_g$.

We also experimented with other design choices, however in our initial experiments this architecture comes out as the best choice. During training both encoder and decoder are trained end-to-end using the joint loss function – *c.f.* Eq. (1). However during testing both encoder and decoder are used in disjoint manner.

**Table 1.** Comparison of bpp, PSNR and SSIM values for different runs of our algorithm on different datasets.

| No. | Cover image | Payload image | No. of epochs | Avg. bpp | Encoder PSNR (db) | Decoder PSNR (db) | Payload % | SSIM encoder | SSIM decoder |
|---|---|---|---|---|---|---|---|---|---|
| 1 | CIFAR10 | MNIST | 50 | 7 | 32.9 | 32.0 | 29.1 | 0.87 | 0.85 |
| 2 | CIFAR10 | CIFAR10 | 50 | 8 | 30.9 | 29.9 | 33.3 | 0.98 | 0.96 |
| 3 | ImageNet | ImageNet | 50 | 8 | 29.6 | 31.3 | 33.3 | 0.88 | 0.88 |
| 4 | ImageNet | ImageNet | 150 | 8 | 32.9 | 36.6 | 33.3 | 0.96 | 0.96 |

**Table 2.** Bpp, PSNR and SSIM values of our ImageNet trained algorithm on different datasets.

| No. | Cover image | Payload image | Avg. bpp | Encoder PSNR (db) | Decoder PSNR (db) | Payload % | SSIM encoder | SSIM decoder |
|---|---|---|---|---|---|---|---|---|
| 1 | PASCAL-VOC12 | PASCAL-VOC12 | 8 | 33.7 | 35.9 | 33.3 | 0.96 | 0.95 |
| 2 | LFW | LFW | 8 | 33.7 | 39.9 | 33.3 | 0.95 | 0.96 |
| 3 | PASCAL-VOC12 | LFW | 8 | 33.8 | 37.7 | 33.3 | 0.96 | 0.95 |

## 3 Experiments and Results

In this section, we report our experimental settings. We also report quantitative and qualitative results of our algorithm on a diverse set of publicly available datasets, that is on ImageNet [1], CIFAR10 [8], MNIST [9], LFW [5] and PASCAL-VOC12 [2].

We randomly divided each dataset sample images into three datasets: training, validation and testing. All the configurations have been done using validation set and we report the final performance on test set.

For payload, we randomly select an image from the corresponding dataset and either convert it to gray-scale or just choose a single channel from the RGB channels. For cover, we randomly select an RGB image from the corresponding dataset.

For all experiments we use the same encoder and decoder architecture as explained in Sect. 2. However each input image is zero-centered. Encoder and decoder weights are randomly initialized using Xavier initialization [3]. For learning these weights we use Adam optimizer with a fixed learning rate of 1E−4 and a batch size of 32 where regularization parameter was set to 0.0001 and $\alpha = \beta = 1$. During each epoch, we disjointly sample images for cover and payload usage from the training set. All the filters in CNN layers are applied with stride of single pixel and using same padding.

We use Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM) index and bits per pixel (bpp) to report the perceptual quality of images produced and embedding capacity of our algorithm.

For our initial experiment, we used cover images ($32 \times 32 \times 3$) from CIFAR10 while payload images were taken ($28 \times 28 \times 1$) from MNIST dataset. For this experiment, we were able to hide approximately 29.1% payload (*i.e.* 7 bpp) in our

cover images with average PSNR of 32.85 db and 32.0 db for encoder and decoder networks produced images respectively – *c.f.* Table 1. These results show that using our algorithm, we can successfully hide a huge payload with reasonably high PSNR and SSIM values. According to our best of knowledge, no one has been able to report such results on this dataset.

However, MNIST is a relatively simple dataset as majority of pixels in each image belong to plain background (white color) class. Thus, we conducted another experiment on CIFAR10 dataset – CIFAR10 being dataset of natural classes contains much larger variation in image foreground and background regions – with identical experimental settings.

In this experiment, both cover ($32 \times 32 \times 3$) and payload images ($32 \times 32 \times 1$) were randomly and disjointly sampled from CIFAR10 training batch. In this experiment we were able to hide a payload of 33.3% (*i.e.* 8 bpp) in our cover images with average PSNR of 30.9 db and 29.9 db for encoder and decoder networks produced images respectively.

From our these experiments, we can conclude that our proposed algorithm is extremely generic and one can, using the same architecture, reliably guarantee huge payloads and acceptable PSNR values for complex images as well – *c.f.* Table 1. For both these experiments we ran our algorithm for 50 epochs.

To further consolidate our findings and to evaluate our algorithm's embedding capacity and reconstruction performance on images of large size, we designed another experiment using ImageNet dataset. A subset of 8,000 images was randomly chosen from one million images. These selected images were then divided into two disjoints sets: training (6,000 images) and testing (2,000 images) – no validation set was used here since we reuse the earlier experiments settings. To allow uniform sized images as cover and payload all of these images were then resized to $300 \times 300$ pixels. For our initial version of this experiment and to ensure a fair comparison with other results, we first ran our algorithm for 50 epochs.

For randomly sampled cover ($300 \times 300 \times 3$) and guest images ($300 \times 300 \times 1$) from our ImageNet test dataset, we were able to hide a payload of 33.3% (*i.e.* 8 bpp) in our cover images with average PSNR of 29.6 db and 31.3 db for encoder and decoder networks produced images respectively. As we were able to hide high payload for similar PSNR values to earlier experiments for this complex dataset as well, so we further explored different experimental settings.

Our final model on ImageNet was trained for 150 epochs further improving the PSNR values for encoder and decoder to 32.92 db (SSIM = 0.96) and 36.58 db (SSIM = 0.96) respectively from 29.6 db and 31.3 db while maintaining similar payload capacity of 33.3% (on average 8 bpp) – *c.f.* Table 1.

To further evaluate the generalization capacity of our algorithm, we ran the ImageNet trained algorithm on sample of 1,000 unseen images from PASCAL-VOC12 [2] and Labelled Faces in Wild (LFW) [5] datasets. Table 2 shows the results of our this experiment. Here, even though our algorithm is trained on different dataset, it is still being able to achieve high payload capacity at high

PSNR and SSIM values which shows the generalization capabilities of our proposed algorithm.

Figure 2 shows a sample of result images from LFW, PASCAL-VOC12 and ImageNet datasets. Here once again we can verify using qualitative analysis that our method is being able to conceal and recover unseen complex payload images.

Therefore, given this quantitative and qualitative analysis, we can conclude that our algorithm is generic and robust to complex backgrounds and variations in objects appearance, thus can be reliably used for image steganography.

**Fig. 2.** Sample results of our algorithm on LFW (top row), PASCAL-VOC12 (middle row) and ImageNet (bottom row) images. In each subfigure, first column represents the cover image $I_h$, second the payload $I_g$, third the hybrid image $O_e$ and fourth column represents the recovered guest image $O_d$.

## 4 Conclusions

In this paper, we have presented a novel CNN based encoder-decoder architecture for image steganography. In comparison to earlier methods, which only consider binary representation as payload our algorithm directly takes an image as payload and uses a pair of encoder-decoder networks to embed and robustly recover it from the cover image. According to our best of knowledge, no such earlier work exists and we are the first one to introduce this method for image-in-image hiding using deep neural networks. We have performed extensive experiments and empirically proven the superiority of our proposed method by showing excellent results with strong payload capacity on a wide range of wild-image datasets.

## References


1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.-F.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
2. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
3. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS, vol. 9, pp. 249–256 (2010)
4. Holub, V., Fridrich, J.: Designing steganographic distortion using directional filters. In: 2012 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 234–239. IEEE (2012)
5. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)
6. Hussain, M., Hussain, M.: A survey of image steganography techniques (2013)
7. Islam, S., Modi, M.R., Gupta, P.: Edge-based image steganography. EURASIP J. Inf. Secur. **2014**(1), 8 (2014)
8. Krizhevsky, A., Nair, V., Hinton, G.: The CIFAR-10 dataset (2014)
9. LeCun, Y., Cortes, C., Burges, C.J.: The MNIST database of handwritten digits (1998)
10. Subhedar, M.S., Mankar, V.H.: Current status and key issues in image steganography: a survey. Comput. Sci. Rev. **13**, 95–113 (2014)