EasyChair Preprint

№ 7705

An improved image steganography model based on Deep Convolutional Neural Networks

Mounir Telli, Mohamed Othmani and Hela Ltifi

April 2, 2022

An improved image steganography model based on Deep Convolutional Neural Networks

Mounir Telli[1,2][0000−0002−3183−5487], Mohamed Othmani[3,4][0000−0001−5617−6062], and Hela Ltifi[5][0000−0003−3953−1135]

[1] National Engineering School of Sfax, University of Sfax, BP 1173, Sfax, Tunisia.
[2] Research Lab: Technology, Energy, and Innovative Materials Lab, Faculty of Sciences of Gafsa, University of Gafsa, Tunisia
telli.mounir@yahoo.fr
[3] Faculty of Sciences of Gafsa, University of Gafsa, BP 2100, Gafsa, Tunisia
[4] Department of Applied Natural Sciences, Applied College, Qassim University, Buraydah, Saudi Arabia
mohamed.othmani@yahoo.fr
[5] Faculty of Sciences and technology of Sidi Bouzid RGIM-Lab ENIS - Sfax
hela.ltifi@ieee.org

Abstract. In this paper, we propose an image steganography model with the use of a DeepCNN grounded autoencoder which allows extracting spatiotemporal features from images. It tends to hide four images in one other image taking into consideration equivalency in terms of size. Thus we try to encode and decode multiple secret images within a single, high-resolution cover image. The quantitative result of this model was arranged using the quantitative index "error per pixel", and the qualitative result was evaluated against the existing approaches.

Keywords: Steganography · Image · spatio-temporal · Deep CNN · auto-encoder

1 Introduction

In information security [11], steganography [12] is an essential technology in the realm of information [13, 11]. Steganography is a technique for encoding secret information (such as a message, a picture, or a sound) into a non-secret object (such as an image, a sound, or a text message) known as a cover object.

The majority of the work in picture steganography [12] has been done to disguise specific content in a cover image. As a result, all known solutions have focused on locating either noisy areas or low-level picture elements such as edges, textures, and so on in the cover image to incorporate as much hidden information as possible without affecting the original image [8].

We offer an amelioration steganography approach for concealing four pictures in one image in this paper. To do this, we created a deep learning network that automatically picks the best attributes from both the cover and secrets images, allowing us to integrate data. The most significant benefit of our method is that

it is general and can be applied to any sort of picture. We aim to make an effort in a similar direction, by utilizing the ideas from the aforementioned papers to encode four images into a single cover image. Our choice is the Convolutional Neural Network as a deep neural network since it's large used nowadays in many domains, including security [12], Object detection and tracking [10, 9], and medicine of neurological ailments [4].

2 Related work

Of several implementations, two are the most aligned and important to our research.

2.1 Hiding Images in Plain Sight: Deep Steganography

Baluja [2] proposed a model in image steganography founded on deep CNN to embed the entire image within another image. This task uses an autoencoder-based deep learning model [1], and validation is done using the weighted total of the reconstruction losses between secret and published secret images also between cover and container images. It has achieved a 100 integration capability. The system's three components are as follows:

- The Preparation Network is responsible for preparing the secret picture to be hidden. If the secret picture (size M*M) is less than the cover image (size N*N), the preparation network gradually grows the size of the secret image until it reaches the size of the cover, dispersing the bits of the secret image overall N*N pixels.
- Hiding Network - generates the Container image using the output of the preparation network and the cover image as input. An N*N pixel field containing depth concatenated RGB channels of the cover picture and modified channels of the hidden image serves as the network's input.
- Reveal Network - utilized by the image receiver;

The picture 1 [2] shows system's three components :

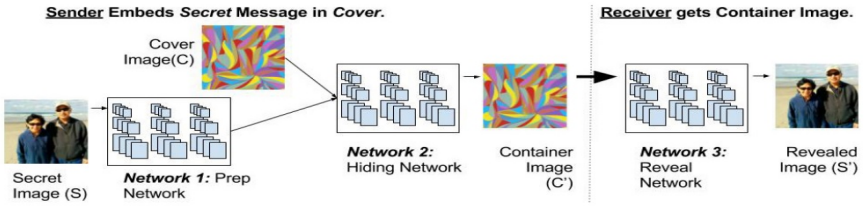

Fig. 1. The three components of the full system. Left: Secret-Image preparation. Center: Hiding the image in the cover image. Right: Uncovering the hidden image with the revealed network; this is trained simultaneously but is used by the receiver.

The architecture and Error propagation of the Baluja system is described in figure 2 [2]. The three networks are trained as a single, large, network. Error term 1 affects only the first two networks. Error term 2 affects all 3. S is the secret image, and C is the cover image.

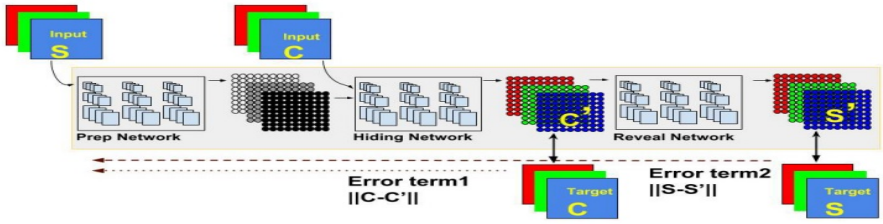

Fig. 2. Baluja system architecture.

Baluja describes how a trained system must learn to compress information from the hidden picture into the least visible sections of the cover image. However, there has been no clear attempt to purposefully disguise the existence of that data from machine detection. The document establishes a standard for encoding single secret images. It does not, however, address multi-image steganography. Figure 3 [2] shows the results of hiding six images, chosen to show varying error rates.

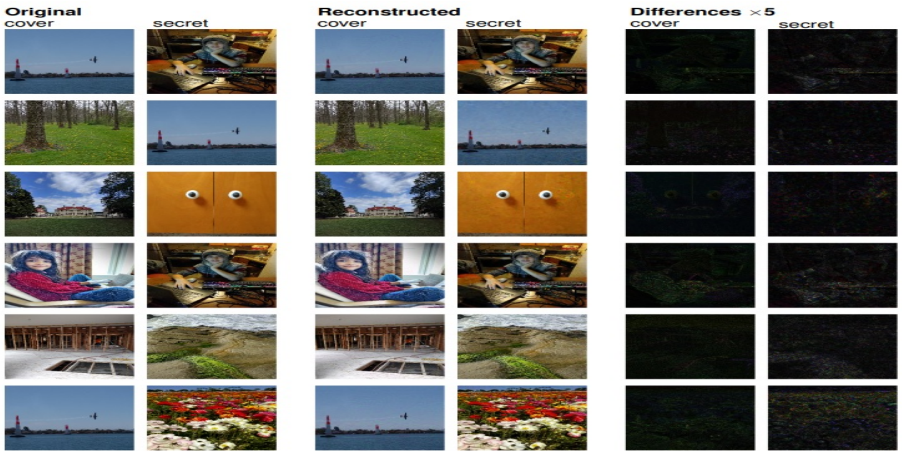

Fig. 3. Six Hiding Results. Left pair of each set: original cover and secret image. Center pair: cover image embedded with the secret image, and the secret image after extraction from the container. Right pair: Residual errors for cover and hidden enhanced 5Œ. The errors per pixel, per channel, are the smallest in the top row: (3.1, 4.5), and the largest in the last (4.5, 7.9).

2.2 Multi-Image Steganography Using Deep Neural Networks

Abhishek [3] used multi-image steganography to conceal three photos in a single cover photo. The hidden pictures included in the code must be retrievable with little loss. The encoded cover picture must be identical to the original. To do so, they integrate the concepts of (Baluja, 2017) and (Kreuk et al., 2019)[6]. They use the notion of a preparation and hiding network as an encoder and a reveal network as a decoder from network implementation [2]. To make this work for numerous pictures, they use the prep network to transmit several secret images, then concatenate the resultant data with the carrier image and send it over the Hiding network. They next use the notion of several decoders, one for each secret picture, to get all of the hidden images from the container image. They expand (Baluja's notion of adding secret pictures with noise in the original cover image instead of putting the secret images at the LSBs of the original cover image to increase the security of our image retrieval model. Figure 4 [6] depicts the model's architecture. Each of the sub-networks basic architecture is as follows:

- Prep Networks: Each prep network is made up of two layers stacked on top of each other. Each layer is made up of three different Conv2D layers. These three Conv2D layers have 50, 10, and 5 channels, respectively, with kernel sizes of 3, 4, and 5 for each layer. Along both axes, the stride length remains constant at one. To preserve the output image in the same dimensions, padding is provided to each Conv2D layer. A ReLU activation follows each Conv2d layer.
- Concealing Network: The hiding network is a five-layer aggregation. The three independent Conv2D layers make up each of these layers. The Conv2D layers in the hidden network have a similar basic structure to the Conv2D levels in the Prep Network.
- Reveal Network: Each of the reveal networks has the same basic architecture as the hidden network, with five levels of Conv2D layers that are comparable in shape.

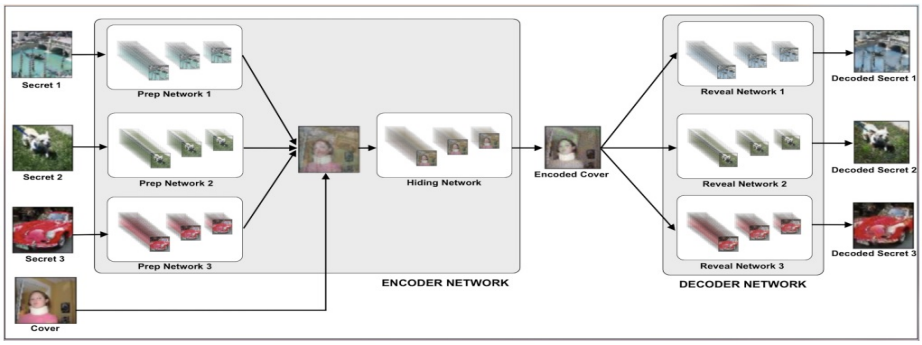

Fig. 4. Abhishek Model Architecture

Figure 5 [6] shows the results of hiding three secret images.

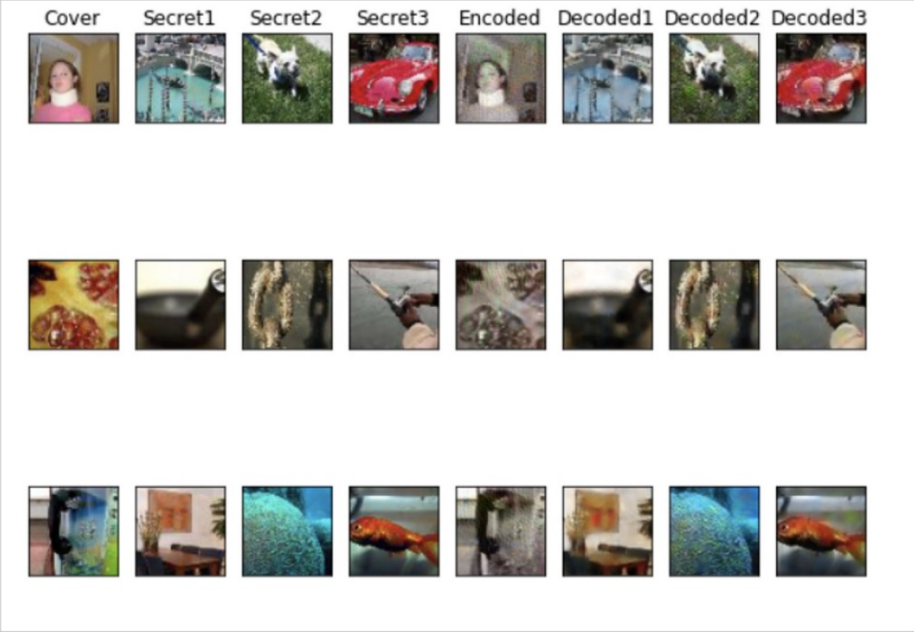

Fig. 5. AResult of hiding three secret images. Left to Right Columns are: Cover Image, Secret Image1, Secret Image2, Secret Image3, Encoded Cover Image, Decoded Secret Image1, Decoded Secret Image2, Decoded Secret Image3.

The loss for all values is projected to rise as the number of photos increases, as more image characteristics are buried in a single image. As a result, they'll need to establish a limit on how many photos may be placed on the cover image to get acceptable results.

3 Proposed Model

In this section, we detail our steganography model consisting of a deep learning-based generic encoder-decoder architecture for image steganography. We then describe the training processes of the proposed model. The overall pipeline of the proposed encoder model is shown in figure 6 and the decoder in figure 7.

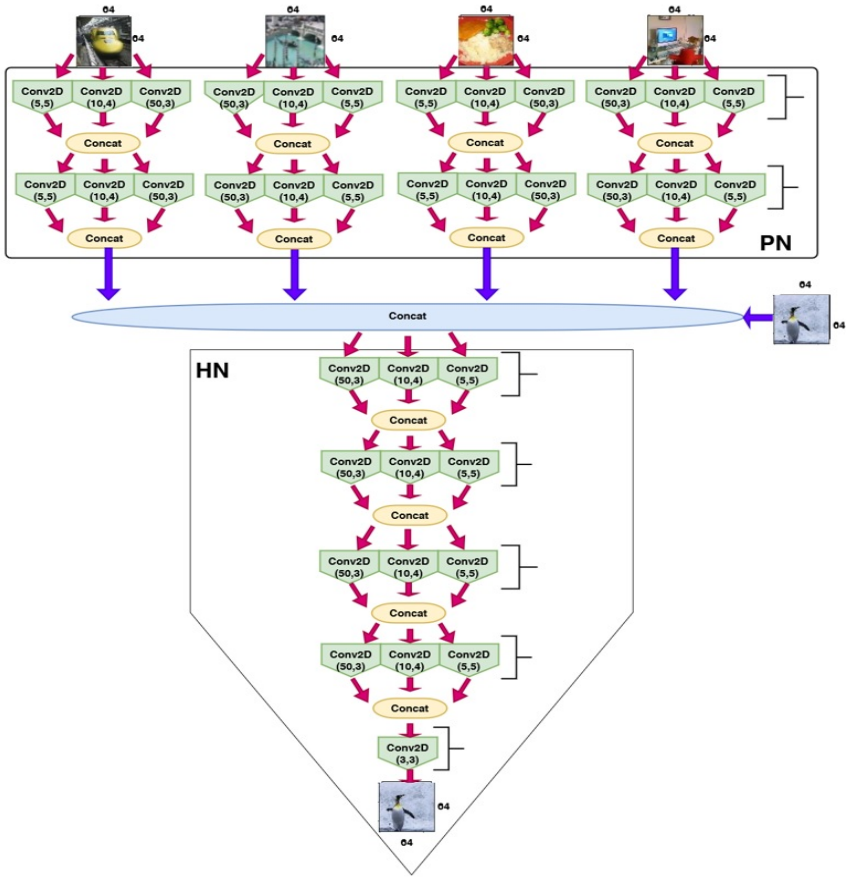

Fig. 6. Pipelene of the Encoder

We aim to combine the model of Baluja [2] with the pipeline of Kumar [7]. Many prominent steganographic approaches for encoding secret communications differ from the suggested steganography framework. Secret pictures are compressed and distributed across all available bits on the cover image using our technology. The decoder and encoder concepts are used in this paper's hiding and extraction networks [5]. The model is composed of three parts: The Preparation Network, Hiding Network (Encoder), and the Reveal Network. Its goal is to be able to encode information about the secrets images S1, S2, S3, and S4 into the cover image C, generating C' that closely resembles C, while still being able to decode information from C' to generate the decoded secrets images S1', S2', S3', and S4' which should resemble the secrets images as closely as possible. The Preparation Network has the responsibility of preparing data from the secret image to be concatenated with the cover image and fed to the Hiding Network.

The Hiding Network then transforms that input into the encoded cover image C'. Finally, the Reveal Network decodes the secrets images S1', S2', S3', and S4' from C'. For stability, we add noise before the Reveal Network. For both the Hiding and Reveal networks, we use 5 layers of 65 filters (50 3x3 filters, 10 4x4 filters, and 5 5x5 filters). For the preparation network, we use only 2 layers with the same structure.

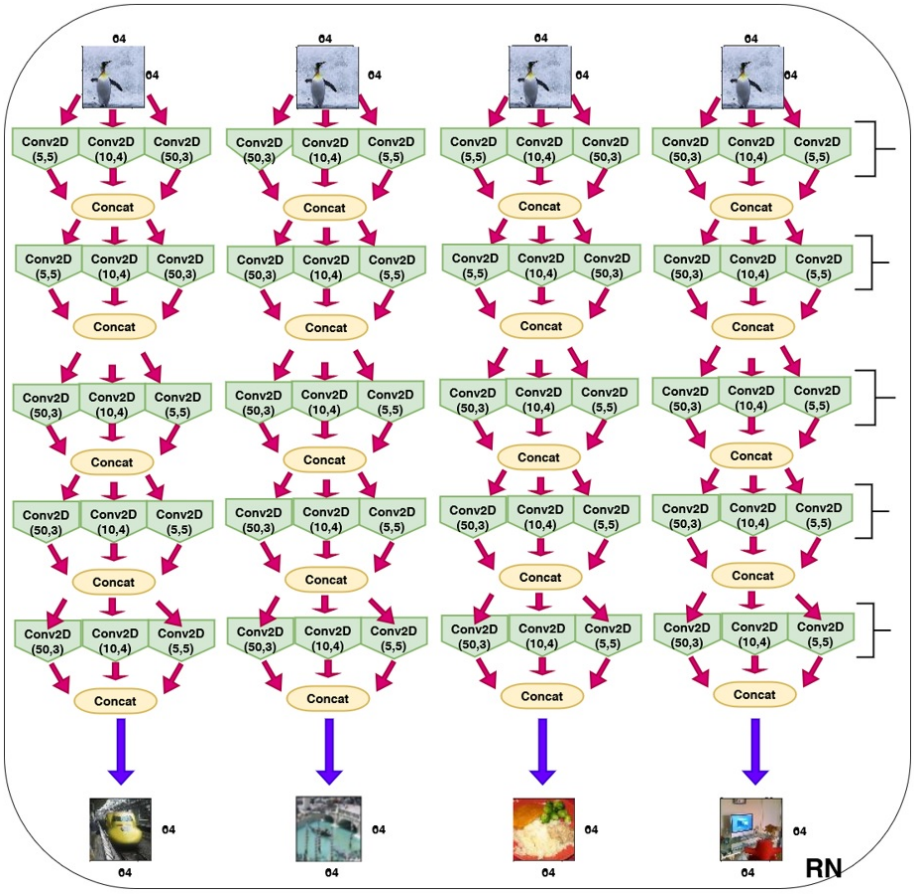

Fig. 7. Pipelene of the Decoder

4 Experimental results

The Dataset we used is Tiny ImageNet Visual Recognition Challenge. Our training set is made of a random subset of images from all 200 classes. Dataset is created by taking 10 images per class for train and 2000 images in total for

train and test. Our experimental aspects and results have been realized on a workstation with an Nvidia Geforce 1650TI GPU card. The model is designed using Python and Conda (Anaconda toolkit) with some dependencies (TensorFlow, NVIDIA CUDA Toolkit, and NVIDIA cuDNN). The training details are explained below: The train set is divided into 2 sections. The First 1600 images are used for training as secret images and the rest 400 for cover images. Adam optimizer has been used. The learning rate remains constant at 0.001 till the first 200 epochs, decreasing to 0.0003 from 200 epochs to 400 epochs and further decreasing by 0.00003 for the remaining iterations. The model has been trained for 1000 epochs with a batch size of 20. Gaussian noise with 0.01 standard deviation is added to the encoders output before passing it through the decoder. The mean sum of squared error has been used for calculating the decoders loss. The loss used is for the full model is represented as:

$$\textbf{Loss} = \|C-C'\|^2+\beta 1*\|S1-S1'\|^2+\beta 2*\|S2-S2'\|^2+\beta 3*\|S3-S3'\|^2+\beta 4*\|S4-S4'\|^2 \quad (1)$$

The training has been performed for value of $\beta 1$, $\beta 2$, $\beta 3$, and $\beta 4$ equal to 1.00. Figure 8 shows the consequences of covering a single cover picture with four hidden images. The encoder/decoder outputs are shown on the left side, while the input pictures are shown on the right. The encoded cover picture resembles

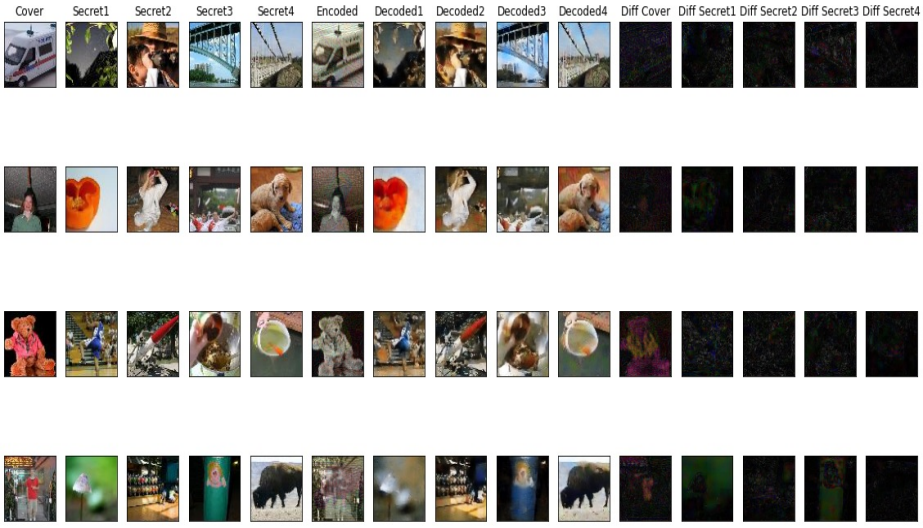

Fig. 8. Result of hiding four secret images. Left to Right Columns are: Cover Image, Secret Image1, Secret Image2, Secret Image3, Secret Image4,Encoded Cover Image, Decoded Secret Image1, Decoded Secret Image2, Decoded Secret Image3, Decoded Secret Image4.

the original cover in appearance but contains no information about the secret

images. In comparison to the scenario where only two secret photos are employed [2], the encoded cover is lossier. In both situations, the hidden photographs are successfully recovered. The mean square error received for our proposed implementation after 1000 epochs were in table I:

Table 1. Values of error per pixel for our proposed model

Image	Proposed results of error per pixel [0, 255]
Secret1 error	19.1207
Secret2 error	17.1059
Secret3 error	18.9291
Secret4 error	13.7798
Cover error	30.6302

The loss curves of the hiding and revealing are computed based on equation 1, where the beta value is (1.00), and the batch size is 20 (represent 4 covers and 16 secrets). The results are shown in figure 9 below:

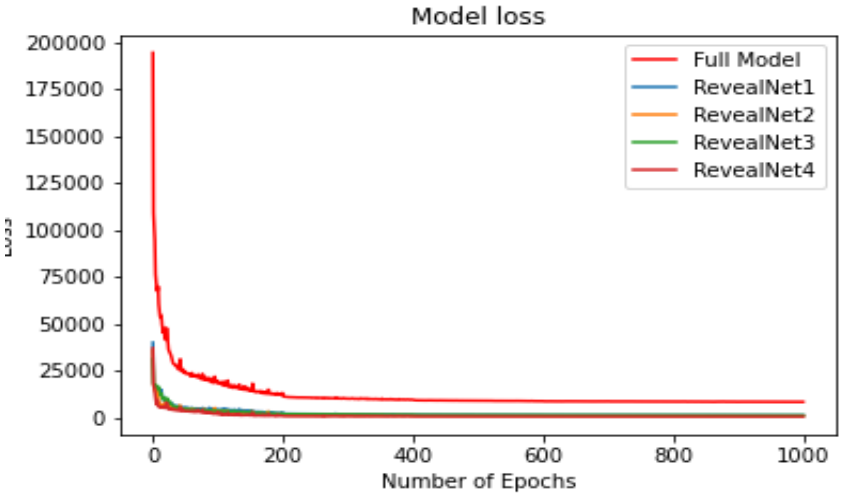

Fig. 9. Loss Curves.

5 Conclusion

In this work, a multi-image steganography method has been designed based on utilized CNN, and the main goal is to achieve improvement in some stego aspects,

including visibility. Our implementation extended the single image steganography model proposed by the recent implementation. We relied heavily on visual perception for overall loss and didnt experiment with various types of losses that could have been better suited for our model. We can use more images or other Datasets to improve our method.

References

1. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. In: Guyon, I., Dror, G., Lemaire, V., Taylor, G., Silver, D. (eds.) Proceedings of ICML Workshop on Unsupervised and Transfer Learning. Proceedings of Machine Learning Research, vol. 27, pp. 37–49. PMLR, Bellevue, Washington, USA (02 Jul 2012)
2. Baluja, S.: Hiding images in plain sight: Deep steganography. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/838e8afb1ca34354ac209f53d90c3a43-Paper.pdf
3. Das, A., Singh Wahi, J., Anand, M., Rana, Y.: Multi-Image Steganography Using Deep Neural Networks. arXiv e-prints arXiv:2101.00350 (Jan 2021)
4. Fourati, J., Othmani, M., Ltifi, H.: A Hybrid Model based on Convolutional Neural Networks and Long Short-term Memory for Rest Tremor Classification. pp. 75–82 (01 2022). https://doi.org/10.5220/0010773600003116
5. Jaiswal, A., Kumar, S., Nigam, A.: En-vstegnet: Video steganography using spatio-temporal feature enhancement with 3d-cnn and hourglass. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2020). https://doi.org/http://doi.org/10.1109/IJCNN48605.2020.9206921
6. Kreuk, F., Adi, Y., Raj, B., Singh, R., Keshet, J.: Hide and speak: Towards deep neural networks for speech steganography (2019). https://doi.org/10.48550/ARXIV.1902.03083, https://arxiv.org/abs/1902.03083
7. Kumar Sharma, D., Chidananda Singh, N., Noola, D.A., Nirmal Doss, A., Sivakumar, J.: A review on various cryptographic techniques and algorithms. Materials Today: Proceedings (2021). https://doi.org/https://doi.org/10.1016/j.matpr.2021.04.583
8. Mielikainen, J.: Lsb matching revisited (2006). https://doi.org/http://doi.org/10.1109/LSP.2006.870357
9. Othmani, M.: A vehicle detection and tracking method for traffic video based on faster R-CNN. Multimedia Tools and Applications (03 2022). https://doi.org/10.1007/s11042-022-12715-4
10. Salah, K.B., Othmani, M., Kherallah, M.: A novel approach for human skin detection using convolutional neural network. The Visual Computer pp. 1–11 (2021). https://doi.org/10.1007/s00371-021-02108-3
11. Singh, L., Singh, A.K., Singh, P.K.: Secure data hiding techniques: a survey. Multimedia Tools and Applications (2020). https://doi.org/10.1007/s11042-018-6407-5
12. Subramanian, N., Elharrouss, O., Al-Maadeed, S., Bouridane, A.: Image steganography: A review of the recent advances. IEEE Access **9**, 23409–23423 (2021). https://doi.org/http://doi.org/10.1109/ACCESS.2021.3053998
13. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. CoRR **abs/1807.09937** (2018), http://arxiv.org/abs/1807.09937