# Neural Steganography and Steganalysis

Abhiraj Chaudhary, Hitesh Goyal, Ashwin Iyer, Sila Shivanandan
*VIT Chennai, Tamil Nadu, India 600127*
*Email: abhiraj.chaudhary2019@vitstudent.ac.in,*
*hitesh.goyal2019@vitstudent.ac.in,*
*ashwiniyer.u2019@vitstudent.ac.in,*
*sila.shivanandan2019@vitstudent.ac.in*

***Abstract*—Steganography is the technique of hiding secret data within an ordinary cover file or message in order to avoid detection. The paper focuses on recognising modern image steganography methods involving Deep Learning and seeing how effective they are against traditional steganography methods. Steganalysis is the procedure of identifying if any form of steganography has been performed on an image. A comparative study between Modern AI Steganography vs Traditional Steganography shall be done while passing them through an AI Steganalysis method.**

## 1. Introduction

Nowadays, the most popular sources of cover objects are digital images. The conventional Steganography methods are now being replaced by more efficient and effective ways. A promising field, Deep Learning, is one of the new methods which is being used to perform Steganography. This paper showcases how effective the recent methods are when compared to other contemporary-older methods.

### 1.1. Problem Statement

Final Goal of the paper is to develop two major neural networks one for steganography and one for steganalysis. Later on, pit them against each other to see which one performs better. In the process, various methods are discovered and worked upon such as DCT, JSteg and MBs.

## 2. Related Work

Conventional Image steganography is done by hiding the information needed to be transmitted by modifying the pixel value (in the spatial domain) or DCT coefficients (in JPEG), such as Jsteg (JPEG steganography), Outguess (guess out steganography), and MBs (Model-based steganography). The new and improved version of F5 (no-shrinkage F5) solves a major shrinkage problem in the F5 algorithm.

### 2.1. MDCFR

For the JPEG domain, the analysis capabilities of Stego are generally extracted directly in the DCT domain for the training of classifiers. Currently, the best manual functionality is called MDCFR (Maximum Residual Cascading Filter Diversity), which can be achieved through predefined cascading filters. While taking the quality factor of the JPEG compression into account, the adaptive weighted mixing of the histogram functions provides the best performance[6].

### 2.2. Methods

After trying to use CNN with different structures for a large number of experiments, SRNet (Deep Residual Network for Steganalysis)[3] introduces a residual structure, which is similar to the concept of residual noise in steganalysis and can be used as a structure for steganalysis extraction. The whole network is composed of three steps: the first segment is responsible for extracting the residual noise, the middle segment must reduce the size of the characteristic graph, and the level of grouping is not added because the Grouping operation will weaken the residual noise extracted in the first segment and the third segment is a standard fully connected layer followed by a linear classifier (Softmax). Among all the integration rates of WOW, SUNIWARD, and HILL, the error rate is usually 1% to 2% lower than that of SCAYeNet technology. Postensemble can further improve detection performance. All in all, these pioneering works show that the performance of the CNN model for steganalysis largely depends on its architecture[11].

### 2.3. Various Methods of Steganography

Some of the conventional methods of Steganography have been referred from a paper[2] and are discussed below.

**2.3.1. Least Significant Bit (LSB).** LSB is one the technique of spatial domain methods. LSB is the simple but susceptible to lossy compression and image manipulations. Some bits are change directly in the image pixel values in hiding the data. Changes in the value of the LSB are imperceptible for human eyes.

**2.3.2. Pixel value Differencing (PVD).** To embedding the data in PVD the two consecutive pixels are selected. Whether the pixels are determine from smooth area or an

edge area. Payload is determined by calculating the difference between two regular pixels.

**2.3.3. The Discrete Fourier Transform (DFT).** Discrete Fourier transform is the transform that are purely discrete: discrete-time signals are converted into discrete number of frequencies. DFT converts a finite list of equally spaced samples of a function into the list of coefficients of a finite combination of complex sinusoids ordered by their frequencies. It can be said to convert the sampled function from its original domain often time or position along a line to the frequency domain. The Discrete Time Fourier transforms uses the discrete time but it converts into the continuous frequency. The algorithm for computing the DFT is very fast on modern computers. This algorithm is known as Fast Fourier Transform i.e. FFT and it produces the same result as of the DFT by using the Inverse Discrete Fourier Transform.

## 2.4. Other Related Work

Boroumand planned a steganalysis detector (SRNet)[2] supported DRN (deep residual network), that achieves smart performance. The model includes four different layers, 2 of which involve the residual layer. The front section is responsible for extracting noise residuals, which is made public by the primary segment, and therefore the middle segment aims to cut back the dimension of the feature graph, the shadow segment, and the last segment, which could be a customarily connected layer, followed by the Softmax linear classifier[9].
The key part of SRNet is that the noise residual extraction section is composed of the first seven layers. Because of the Avg. pooling operation could be a low-pass filter, it enhances content and suppresses noise like stego signals by averaging embedding changes. Pooling operations aren't value-added to the highest seven layers of the model[5].
Their work shows that the residual structure is extremely helpful in steganalysis. It can't solely capture the advanced applied math information of digital pictures and enhance the knowledge from secret messages, however conjointly solve the gradient dispersion development in deep convolution neural networks, but this might not be the most effective residual layer structure combination for steganalysis tasks[8].

# 3. Method

## 3.1. Datasets

**3.1.1. ALASKA-2.** The ALASKA-2 dataset is one of the important datasets which is used in Steganalysis. It contains 5 folders: Cover, JMiPOD, JUNIWARD, UERD, Test.

- Cover contains 75k unaltered images meant to use in training.
- JMiPOD contains 75k examples of the JMiPOD algorithm applied to the Cover images.
- JUNIWARD contains 75k examples of the JUNIWARD algorithm applied to the Cover images.
- UERD contains 75k examples of the UERD algorithm applied to the Cover images.
- Test contains 5k test set images which will be used for predicting.

**3.1.2. Linneaus 5.** This is an image dataset[5] which was chosen to test the different steganography methods which needed to be custom made. Some attributes of this dataset are:

- 5 classes: berry, bird, dog, flower, other (negative set)
- Images are 256x256 pixels, color (downsampled versions: 128X128, 64X64 and 32X32 pixels).
- 1200 training images, 400 test images per class.
- Images were downloaded from pixabay.com.

Although, to the relevance of the paper, the only important thing is that the sizes of the images are appropriate and can be used for steganography and subsequent steganalysis.

**3.1.3. Custom Datasets.** The ML Steganalysis Detection dataset is a simple dataset, with 500 stego images and 500 normal images. The objective is to build a classifier to predict given an image whether there is a secret image hidden inside it. The Steganalysis Algorithm Classification dataset consists of images which have secret images embedded in them using 4 algorithms:

1) JUNIWARD
2) JMiPOD
3) MLStego
4) UERD

The objective of the dataset is to predict which algorithm has been used to hide the image.

## 3.2. Steganography

**3.2.1. Method.** To perform steganography, a paper, Deep Steganography[6] was used as reference. It was accompanied by an implementation which needed some minor changes for it to work on the Linneaus 5 dataset. After a working implementation was attained, certain changes were made in the model to improve the performance. After doing that, the Linneaus 5 images were used to generate stego images. These images were then passed into the steganalysis model to see how identifiable they are.

**3.2.2. Architecture.** The architecture used in the Deep Steganography paper was not changed much and was mostly used as it is. The loss function was also not changed and the Squared Error along with a coefficient for secret image were used to train the network. The original implementation had a Gaussian Noise layer which hindered with the performance of the network and it was removed to get more idealistic results.

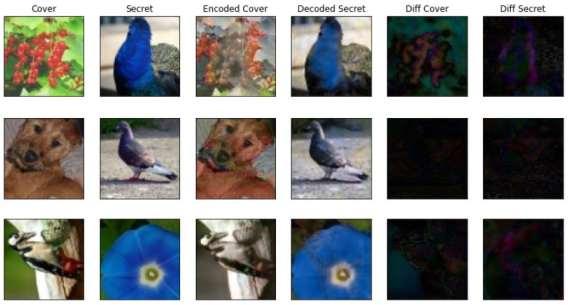


Figure 1. Implementation of Deep Steganography.

### 3.3. Steganalysis

For Steganalysis, we will be using a technique called Knaive Method. In this method, we will flatten the RGB images (both stego and normal) into a vector and the find the cosine dissimilarity between the two vectors. If the images are alike, then cosine dissimilarity(1 - similarity) must be 1, else it will be less than 1 and we will know that there is some information in the image. After achieving the dissimilarities, we use Softmax function to get the values between 0-1 for easier calculation[15].
We use CNN as it follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed. For Steganalysis, we use a CNN model with 5 layers, Max Pooling and Batch Normalization, with Adam as optimizer, learning rate of 0.001 and train it for 10 epochs and batch size of 100.

### 3.4. Combining the methods

To combine the two methods, a custom dataset has been created which has been mentioned previously. A part of Alaska 2 has been combined with the stego predictions of Deep Steganography implementation. Then, the Steganalysis network has been trained to classify the method of steganalysis used to see how well the network is able to classify between the steganography methods. To make the model more robust and get clearer results, non-stego images are also added to the dataset.

### 3.5. Results

The results attained suggest that the Neural Network is able to correctly classify the four classes with the following accuracies:

1) JUNIWARD - 75%
2) UERD - 87%
3) JMiPOD - 77%
4) MLStego - 100%*

These accuracy scores suggest that the individual steganography algorithms can quite easily be detected by Neural Networks. The overall accuracy of the network to classify the different forms of Steganography is 55% which suggests that it is harder to classify the steganography algorithm even though steganography is easily detected.
*Regarding the MLStego accuracy, the model actually predicts all images as stego, suggesting it is unable to identify the pattern and predicts everything as the same.

## 4. Conclusion

Looking at the results, it can fairly be concluded that the Steganalysis method is able identify and detect Steganography but it finds it hard to classify it into the different types. Also, the model is unable to detect the Deep Steganography which suggests that the ML based methods are more effective than the other conventional methods. It can be safe to say that AI can be a tough contender among the various methods of steganography invented.

## 5. Future Work

### 5.1. Steganography

The results make it clear that the Deep Learning based steganography has scope of improvement. The implementations suggest that a modification of the Network architecture and usage of a more appropriate loss function can be effective in making the steganography even harder to notice or predict. Another change which can be introduced is cover selection based on the given secret. If the secret and cover have a low pixel difference, hiding one in the other can become even harder to identify.

### 5.2. Steganalysis

The Steganalysis model can be tuned, tweaked and modified for better performance. More recent approaches like hyper-parameter tuning, Transfer Learning can be explored to see the if there can be any improvement. Some modification of preprocessing techniques can make it easier for the model to identify stego patterns. Also, ensemble models which can first detect the presence of steganography and then go on to predict the method can be used.

## 6. Acknowledgement

This research work was supported by Vellore Institute of Technology, Chennai under Professor Rajesh Kumar. We would like to thank him deeply for his support and motivation which helped us create this paper and perform research on this topic.

## 7. References

1) Bas, P., Filler, T., Pevnỳ, T.: “break our steganographic system”: The ins and outs of organizing a boss. In: International workshop on information hiding, pp. 59–70. Springer (2011)

2) Kaur, Harpreet, and Jyoti Rani. "A Survey on different techniques of steganography." MATEC Web of Conferences. Vol. 57. EDP Sciences, 2016.
3) Das, Abhishek, et al. "Multi-Image Steganography Using Deep Neural Networks." arXiv preprint arXiv:2101.00350 (2021)
4) Boroumand, M., Chen, M., Fridrich, J.: Deep residual network for steganalysis of digital images. IEEE Transactions on Information Forensics and Security 14(5), 1181–1193 (2018)
5) Denemark, T., Sedighi, V., Holub, V., Cogranne, R., Fridrich, J.: Selection-channel-aware rich model for steganalysis of digital images. In: 2014 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 48–53. IEEE (2014)
6) Chaladze, G. Kalatozishvili L. 2017. Linnaeus 5 Dataset for Machine Learning.
7) Baluja, Shumeet. "Hiding images in plain sight: Deep steganography." Advances in Neural Information Processing Systems 30 (2017): 2069-2079.
8) Fan, L., Sun, W., Feng, G.: Image steganalysis via random subspace fisher linear discriminant vector functional link network and feature mapping. Mobile Networks & Applications (2019)
9) Feng, G., Zhang, X., Ren, Y., Qian, Z., Li, S.: Diversity-based cascade filters for jpeg steganalysis. IEEE Transactions on Circuits and Systems for Video Technology 30(2), 376–386 (2020)
10) Fridrich, J., Goljan, M., Hogea, D.: Steganalysis of jpeg images: Breaking the f5 algorithm. In: International Workshop on Information Hiding, pp. 310–323. Springer (2002)
11) Fridrich, J., Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security 7(3), 868–882 (2012)
12) Fridrich, J., Pevnỳ, T., Kodovskỳ, J.: Statistically undetectable jpeg steganography: dead ends, challenges, and opportunities. In: Proceedings of the 9th workshop on Multimedia & security, pp. 3–14 (2007)
13) Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
14) Guo, L., Ni, J., Shi, Y.Q.: Uniform embedding for efficient jpeg steganography. IEEE transactions on Information Forensics and Security 9(5), 814–825 (2014)
15) https://www.kaggle.com/tanulsingh077/steganalysis-a-knaive-approach(Steganalysis)