Image Steganography using Deep Learning

Dinesh Vennapoosa (24002307784106), Sandeep Sirivuri (24002304289125), Maddula Krishna Sai Prakash (2400230785964231)
Luddy School of Informatics, Computing and Engineering

1 Introduction

In recent times we can witness numerous instances of cybercrimes related to "deep fakes". Deep fake refers to the application of sophisticated artificial intelligence and machine learning techniques to create fake images, text-based data, etc. "Image steganography" can be an effective tool against Deep fake-based cybercrimes [1]. Image steganography is the technique of hiding secret information i.e., an image or text in another image without changing the appearance and quality of the image. This technique also has various applications in data security, where sensitive information must be transmitted without arousing suspicion. In our project, our aim is to implement Image in Image and Text in Image steganography through an encoder-decoder architecture that can effectively hide secret information within a cover image. Our method enables simultaneous concealment and retrieval of secret images within cover images, while maintaining imperceptibility to human vision.

2 Problem Description

Even though Deep Learning has been successful in various computer vision tasks such as image recognition, object detection, and semantic segmentation, the application of deep learning in steganography is still in its nascent stage, and further research is needed in this field. Our project aims to implement image steganography and explore the potential of deep learning in image steganography and develop a model that can hide information(image/text) within an image while maintaining its quality and avoiding detection by other retrieval techniques.

3 Dataset Description

We have considered two datasets tiny imagenet dataset [2] and text data [3] for our project. Tiny ImageNet dataset for images dataset and text files dataset from mendeley.com consisting of 99 text files. Tiny image net consists of images of resolution 64 x 64.We have considered 500 cover images and 500 secret images from Tiny ImageNet for Image in Image stegenography task and 99 cover images from Tiny ImageNet dataset and 99 text files from above mentioned source for text in image steganography task.

4 Methodology

We have considered two tasks for implementation namely Image in Image Steganography and Text in Image Steganography. Image in Image steganography task involves hiding one or more images within another cover image without distorting the appearance of the cover image.Text in Image Steganography task involves hiding text within another cover image without distorting the appearance of the cover image

4.1 Data preprocessing

In the case of Image in Image steganography, the images are first resized to a standard shape of (64,64,3) and then the pixel values are normalized by dividing them with 255. For Text in Image steganography, in addition to the above image preprocessing steps, text preprocessing is also performed. This includes converting the text to lowercase, removing any non-alphabetic characters, and tokenizing the text. The text is then padded to ensure that all messages are of equal length, and finally, one-hot encoding is applied to represent the text in a numeric format. This results in a final text representation with a shape of (18, 25), where 18 is the maximum length of the text, and 25 is the number of unique characters. These preprocessing steps are essential to ensure that the input data is standardized and suitable for use in the model training process.

4.2 Steganography technique

Steganography is a technique that involves hiding secret information in plain sight. There are several methods available for performing steganography, including Least Significant Bit (LSB) replacement, Spatial Domain Modification, Transform Domain Steganography, and Deep Learning-based Steganography using CNNs (Convolutional Neural Network). In this work, we have utilized a Deep Learning-based approach, specifically a CNN-based model, for hiding the secret information within an image. The implementation details are described in Section 4.3, and the results are compared with those obtained using the LSB method, by using the Structural Similarity Index (SSIM) as the evaluation metric.

4.3 Deep Learning Architecture

4.3.1 Image in Image Steganography workflow

The proposed algorithm for image-based steganography consists of three main components illustrated in Figure 1: a preparation network, an encoder, and a decoder. The preparation network is composed of 6 convolutional layers, which are used to extract different features of the secret image, such as edges, borders, colors, and complex structures. The output of the preparation network is a tensor that represents the features of the secret image. The encoder is a neural network that takes in the tensor output of the preparation network and concatenates it with a tensor of the cover image that undergoes 15 convolution layers in the encoder. The cover image tensor serves as a carrier for the secret image tensor, and the encoder outputs a Stego image tensor that contains both the cover and secret images. The purpose of the encoder is to blend the features of the cover and secret images in a way that the resulting stego image appears similar to the original cover image, while also concealing the secret image within it.

The decoder, or reveal network, is another set of 15 convolutional layers that are used to extract the features of the stego image tensor. The reveal network reconstructs the features of the secret image from the stego image tensor and outputs a reveal image tensor that closely resembles the original secret image. The purpose of the reveal network is to extract the hidden information from the stego image and reconstruct the original secret image as accurately as possible.

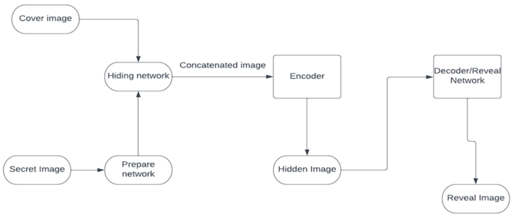

Figure 1: Encoder and Decoder architecture for Image in Image steganography

4.3.2 Text in Image Steganography workflow

The text in image steganography algorithm as shown in Figure 2 is similar to that of image in image steganography, but with a focus on the hidden text message rather than an entire image. The algorithm starts by converting the text message into a tensor representation, which is then concatenated with a tensor of the cover image to form a stego image. The encoder network, which consists of 3 convolutional layers, takes in the concatenated tensor and produces a stego image tensor. The reveal network, on the other hand, is composed of 2 convolutional layers that are used to extract the hidden text message from the stego image tensor. By using convolution layers in both the encoder and reveal networks, the algorithm achieves a high level of accuracy and efficiency in hiding text in an image. However, extracting the hidden text message has a low accuracy rate, which is a known limitation of this technique.

4.4 Loss function

Our steganography algorithm utilizes two mean squared error loss functions. The encoder loss measures the difference between the cover image and the stego image, while the decoder loss measures the difference

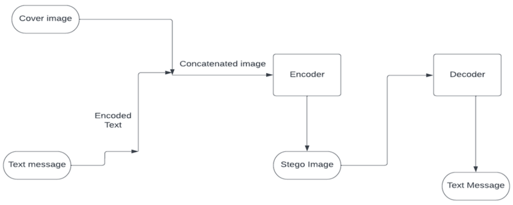

Figure 2: Encoder and Decoder architecture for text in image steganography

between the original secret image (in image steganography) or one hot encoded vector of the secret text (in text in image steganography) and the revealed image or text. These loss functions help to optimize the algorithm by ensuring that the stego image closely resembles the original cover image and the hidden information is accurately extracted from it.

4.5 Evaluation

4.5.1 Image in Image steganography

To evaluate the performance of our Image in Image steganography algorithm, we utilize the Structural Similarity Index (SSIM). This index measures the similarity between the reveal image tensor and the original secret image tensor, as well as between the cover image tensor and the stego image tensor output from the encoder. The SSIM takes into account various factors such as structure, luminance, and contrast, and a higher SSIM value indicates a more successful steganography algorithm.

4.5.2 Text in Image steganography

In the case of Text in Image steganography, we use the SSIM to compare the stego image output from the encoder with the original cover image, in order to ensure the quality of the cover image used for steganography. The higher the SSIM value, the higher the degree of similarity between the stego image and the original cover image, indicating a better-quality steganography algorithm. In addition to evaluating the performance of our proposed steganography algorithm, we also compare it with the Least Significant Bit (LSB) replacement method as a baseline. We use the SSIM values of the stego images outputted from the encoder to compare with the results of stego images obtained using the LSB replacement method, to determine the quality of our steganography algorithm in both Image in Image and Text in Image steganography.

4.6 Results

For the evaluation of steganographic models, we consider the structural similarity index measurement (SSIM) score as an evaluation metric which measures structural, luminance, and contrast similarity between two images.

4.6.1 Image in Image steganography

we can observe the similarity between the cover image and hidden image (stego image) as well as the secret image and revealed image in the Figure 3.

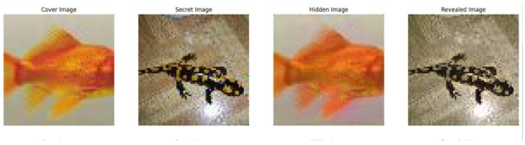

Figure 3: Result of image in image steganography

In table 1 and table 2, we can observe the SSIM scores for the encoder stego image, LSB stego image, with Cover image and Decoder /reveal image with secret image. Even though the traditional Least Significant Bit(LSB) steganographic method has higher SSIM , it's not recommended to use the LSB technique as it is not secure since information can be easily extracted from images using that technique and our method using deep learning which is more secure. So we are using LSB as a baseline to evaluate our deep learning steganography models. We can observe that SSIM scores for Encoder stego image and Decoder reveal image are reasonably good indicating that the image quality has been preserved while hiding the image within an image.

Table 1: SSIM Score of Encoder and LSB Stego Images

Cover Image ID	Encoder Stego Image Score	LSB Stego Image Score
0	0.80	0.99
1	0.79	0.99
2	0.95	0.99
3	0.89	0.99

Table 2: SSIM Score of Decoder/Reveal Image

Secret Image ID	Decoder/Reveal Image Score
0	0.95
1	0.81
2	0.96
3	0.92

4.6.2 Text in Image steganography

we can observe the similarity between cover images and Stego images which contain the hidden text in the Figure 4.

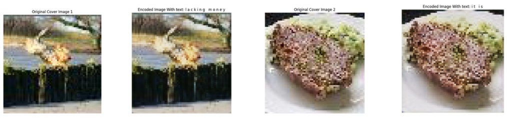
Original Cover Image 1
Encoded Image With text: lacking money
Original Cover Image 2
Encoded Image With text: it is

Figure 4: Encoder and Decoder architecture for text in image steganography

The table 3 shows the SSIM score along three channels of the image. We can observe that the deep learning steganography technique has preserved the image quality while hiding text within the image as indicated by high SSIM scores. We can also observe that the results obtained are similar to the traditional Least Significant Bit(LSB) steganographic method.

Table 3: SSIM Score of Encoder and LSB Stego Images

Cover Image ID	Encoder Stego Image Score	LSB Stego Image Score
1	0.96712426 ,0.97133689, 0.94409444	0.99999178, 0.99999342, 0.99999409
2	0.97162895, 0.95166751 0.88214915	0.9999624 ,0.99995595, 0.99995717

5 Conclusion

In the current era of increasing cybercrimes associated with deepfakes, employing image steganography as a countermeasure will be very effective. Through our project, we have successfully implemented an encoder-decoder architecture for image steganography that can effectively conceal secret information, such as images or text, within a cover image. Our approach demonstrates the potential of deep learning techniques in the field of image steganography, allowing for secure transmission of sensitive information without attracting attention or compromising image quality also at the same time allows us to distinguish between fakes image and authentic images as we can hide text or an image as a unique identifier within another image as a safety distinguishable feature for identification.

6 Future Work

These methods can also be applied to audio and video files, as they offer significant potential for enhancing the security of communication channels. There is ample opportunity for advancements in deciphering concealed data and developing innovative encoding techniques for concealing data for securing our audio, video, text, and image files.

References

[1] V. Himthani, V. S. Dhaka, M. Kaur, G. Rani, M. Oza, and H. N. Lee, "Comparative performance assessment of deep learning based image steganography techniques," *Scientific Reports 2022 12:1*, vol. 12, pp. 1–16, 10 2022. [Online]. Available: https://www.nature.com/articles/s41598-022-17362-1

[2] "Stanford university cs231n: Deep learning for computer vision." [Online]. Available: http://cs231n.stanford.edu/tny-imagenet-200.zip

[3] Q. Woo, "Stego images," vol. 1, 2018.

Appendices

A Contribution of each Member

- **Mr. Maddula Krishna Sai Prakash** Problem statement, architecture development, CNN implementation, report preparation.
- **Mr. Sandeep Sirivuri** Literature review, architecture development, CNN implementation, Evaluation metrics.
- **Mr. Dinesh Vennapoosa** Architecture development, CNN implementation, Report preparation, Other Model comparison.

Github: `https://github.com/Dini-49149/Image_steganography_DL`