DeepSteg: A Deep Learning approach to steganography detection

EE6310 Final project report

Paper ID 14

Abstract

Steganography is a technique of hiding secret information within a non secret file to maintain confidentiality and security. It has become a popular tool for ensuring data security. In particular, Image steganography has gained popularity due to the widespread use of digital images for communication. It involves hiding secret information within the pixel values of an image, making it imperceptible to the human eye. Steganalysis is the process of detecting and decoding hidden messages within a file, which has become increasingly challenging with the advent of advanced steganography techniques. Deep learning, being a powerful tool in various applications, has also shown immense potential in the field of steganography. This paper intends to explore various deep learning methods for detecting and decoding steg Images.

1. Introduction

The goal of Steganography is to ensure that the hidden information remains undetected by unauthorized parties. It is the intersection of several domains, including cryptography, information theory, and machine learning. Steganography can be broadly classified into two types: spatial domain steganography and frequency domain steganography. Spatial domain steganography involves modifying the pixel values of the cover image, whereas frequency domain steganography involves modifying the frequency components of the cover image. The choice of steganography method depends on the nature of the cover media and the intended application. And based on the carrier file used they can be classified as- Image steganography, Text steganography, Video steganography, and Audio steganography.

In this paper, we focus on Image steganography, where secret information is hidden within the pixel values of an image. Our research aims to explore various Deep learning techniques for detecting and decoding steganographic images. Specifically, we focus on the effectiveness of deep learning-based methods for steganalysis and compare their performance to traditional steganalysis methods.

1.1. Image Steganography

Image steganography is a technique of hiding secret messages or data within an image without changing the appearance or quality of the image. Its goal is to keep the hidden message undetectable by human or machine analysis.The process involves embedding the secret message or data into the the pixels of the image. The hidden message can be extracted from the image using a steganography decoder. This decoder reverses the embedding process to extract the hidden message or data.

There are several techniques involved in image steganography, including LSB (Least Significant Bit) Insertion, Masking and Filtering, DCT (Discrete Cosine Transform), discrete wavelet transform (DWT), or singular value decomposition (SVD) to embed secret information in the cover image.

Image steganography can be used for various purposes, including secret communication, copyright protection, and digital forensics. However, it can also be used for malicious purposes such as hiding malware or other harmful content.

1.2. Deep Learning

Deep learning techniques used for image steganography can be broadly divided into - traditional methods, Convolutional Neural Network-based and General Adversarial Network-based methods. They has played a significant role in enhancing the security and robustness of steganographic techniques. These techniques can be used for:

- **Steganography techniques:** DL techniques can be used to make the traditional methods more secure and robust such as Generative adversarial networks (GANs) have been used to create realistic-looking images that can be used as covers for hidden messages. Additionally, autoencoders have been used to encode and decode hidden messages, improving the efficiency and reliability of steganographic techniques.
- **Steganalysis:** DL algorithms can be used to develop powerful steganalysis tools in digital media. Such as Convolutional Neural Networks (CNNs) have been trained to identify steganographic patterns in images

and videos, making it more difficult for attackers to hide messages undetected.

Deep learning techniques are important in ensuring the security and data privacy of steganography methods.

2. Literature Review

2.1. Steganographic Methods

The three primary categories of steganographic techniques are traditional image steganography, CNN-based image steganography, and GAN-based image steganography. Traditional methods employ techniques that are not related to machine learning or deep learning algorithms.

Many traditional methods [4] [13] are based on the Least Significant Bits (LSB) substitution, which involves hiding the information within the least significant bits of the pixel values of an image. This is possible because changing the least significant bits does not usually result in noticeable changes to the overall appearance of the image. To perform LSB substitution, the secret information is first converted into binary form. Then, the cover image is scanned to determine the least significant bits in the noisy area. The binary bits from the secret image are then substituted in the LSBs of the cover image.

CNN-based Image steganography models typically follow an encoder-decoder architecture. This involves taking two inputs - a cover image and a secret image and passing them through the encoder to generate a stego image. The stego image is then passed through the decoder to extract the embedded secret image. The goal is to train the network to embed the secret information while minimizing the perceptible impact on the cover image.

GAN-based Image steganography methods use Generative Adversarial Networks (GANs) to hide secret information within an image. GANs are a type of deep CNN that use an adversarial process to generate images. In a GAN, two networks - a generator and a discriminator compete against each other to generate realistic images.In GAN-based steganography methods, the generator network is trained to embed the secret information into the cover image, while the discriminator network tries to detect the presence of hidden information.

2.2. Traditional steganalysis methods

Traditional Steganalysis methods are often employed to detect steganographic techniques that use basic algorithms, such as Least Significant Bit (LSB) substitution or a variant of LSB method. Traditional steganalysis methods attempt to detect the modifications by analyzing the statistical properties of the cover image, the algorithm looks for changes in the distribution of pixel values or the frequency of occurrence of certain patterns so that it finds the changes in the overall structure of the image.

While traditional steganalysis methods can be effective at detecting simple steganographic techniques, they may not be as effective against more advanced methods that use Machine Learning techniques

2.3. Deep Learning Steganalysis Methods

[8]Earlier for doing steganalysis using deep learning there used to be many methods. One such method was Spatial Rich Model (SRM) features of the spatial domain and by Kodovsky [7]. SRM features are extracted in the form of 106 submodels, each submodel with approximately 300 features,giving a total of 34671 features. The various variants of SRM such as Projection Spatial Rich Model by Holub and Fridrich and others are also used for providing better results.The manual designing of such high-dimensional features followed by extraction is quite challenging. Moreover, it burdens the classification task in universal steganalysis. This redirected the researchers towards finding a way of automatically getting out of useful features from images through deep learning. Deep learning (DL) has already proven its potential in various other fields as computer vision, pattern recognition, and image classification.

2.3.1 Why we use CNNs

[16]The noise residuals from each image are calculated using a total of 39 filters. The researchers using SRM manually design the filters. Each filter predicts the core pixel based on the nearby pixels. The residual is then created by deducting the anticipated value from the pixel's original value. This method, which is comparable to convolution, uses 39 of these filters to calculate the residuals for each and every pixel. i.e., figuring out the residuals. [7] The calculation of pointwise sigmoid nonlinearities in the CNN convolution layer, followed by the average/max pooling and subsampling layer, may be analogous to the steps that were taken. The SRM has a drawback in that it is a manual process, whereas CNNs automatically learn from training.

2.3.2 Using CNNs in Spatial Domain

[3] [14]CNNs can be trained to identify changes in pixel values that are not visible to the human eye. This can be done by training the CNN to identify patterns in the image that are not present in the original image.There are two domains in which we can apply the CNNs. first is spatial domain another one is transform domain. In this section we discuss about applying CNNs in spatial domain. So this method what we can do is while doing the steganalysis first we process the image using a image processing layer and then pass it to the CNN layer. So that the values initialsied in the CNN layer are similar to one in the SRM process.This we do it for getting the similar or better results.Since providing cover and stego images directly makes CNN con-

vergence challenging in deep learning as well, noise residuals of both cover and stego images are provided as input to the initial convolution layer, and subsequent features are learned on these noise residuals to enable the identification of a distinct difference between cover and stego images. As a result, an image processing layer that comes before each CNN convokes input images through several filters to determine these residuals.The activation functions used in CNN are sigmoid and different forms of it, but researchers have provided Gaussian Neuron (GN) activation is giving more good results.It provided the best results.

2.3.3 Pretrained Networks

[3]After doing a lot of research, researchers has found that the best thing to use is the pretrained networks for doing the steganalysis. Becuase those have been trained on millions and millions of datasets so they outperform in any of the normally trained CNNs.Initially, they were created from scratch in the field of steganalysis but later moved to highly efficient pretrained networks such as SRNet, ResNet and EfficientNet and found significant improvement in results on more challenging datasets such as ALASKA-I and ALASKA-II. Which gave them better results for the steganalysis.Pretrianed networks always had improved results.

3. Further literature review

Steganographic schemes in spatial domain can be categorized into LSB embeddings/replacement and Cost function based embedding, Adaptive embedding and DL based techiniques to utilize the robustness of CNN and GAN. To widen our understanding we dig deeper into the LSB, DCT (Discrete Cosine Transform) [2], J-UNIWARD [17], UERD [6](Uniform Embedding Revisited) techniques which are used for the image steganography.

3.1. Working of LSB steganography

LSB uses the fact that out human visuaol system is imperceptable of small changes to a pixel of an image. The LSB of an image of selected pixels is replaced, But the selection of pixels may be successive selection or pseudo random.

3.1.1 Design Details

[10]This section focuses on algorithms of LSB Steganography and Steganalysis

Algorithm for Hiding (Steganography)

1. Read the original image and the image which is to be hidden in the original image
2. Shift the image to hide in the cover image by X bits
3. And the original image or cover image with 240 which is 11110000 So four MSb's set to 0.Because of this only four LSB's considered
4. The shifted hidden image and the result of step 3 are bitored. This makes changes only in the X Lsb bits so that the image is hidden in the original image further.

This image can be called as the stego image. As [11] suggestes that in LSB there is High Percentage Distortion less resultant image . And here the image manipulation is also low. But [11] also says that the LSB introduces noise to image.Even though the paper suggests the steganlysis detection is low. More deeplearning ways have proven that it could be done easily and other methods chi-square attack, the histogram analysis, and the RS analysis have proven to be more effective in doing the steganaylsis.And attackers can attack easily to get the steganographic data. We found that LSB steganography is not so good in hiding the secret messages.

3.2. Working of DCT steganography

[2]Discrete Cosine Transform (DCT) is a widely used technique in image steganography, which involves hiding secret data within digital images without visibly altering the image quality. The basic idea behind this technique is to transform the image into the frequency domain from spatial domain using DCT, and then embed the secret data into the lower frequency coefficients, which are generally more significant and less sensitive to noise than the higher frequency coefficients. The frequency coefficients of the image can be modified to embed the secret data.
In image steganography, the secret data is usually embedded by modifying the DCT coefficients of the image. The least significant bit (LSB) of the coefficients can be replaced with the secret data bits, or the higher frequency coefficients can be modified without significantly altering the image quality.

To extract the secret data from the stego-image, the same process is applied in reverse: the image is transformed using DCT, the lower frequency coefficients are extracted, and their values are compared to a threshold value to determine the value of the embedded bit.The LSB replacement can be done using a secret key to ensure the security of the hidden message.

DCT-based steganography is widely used because it can provide a high embedding capacity while maintaining good image quality and low detectability. However, careful consideration should be given to the amount of data to be hidden and the compression algorithms used to ensure the security and reliability of the hidden message.

3.3. Working of J-UNIWARD

[17]J-uniward steganography works by dividing the JPEG image into blocks of pixels and modifying the dis-

EE6310 #14

crete cosine transform (DCT) [2] coefficients of these blocks to encode the hidden message. The modification process is controlled by a secret key that is known only to the sender and receiver of the message. J-uniward steganography is considered more secure than some other steganography techniques, it is not foolproof and can still be detected by sophisticated statistical analysis or by using specialized steganalysis tools.

4. Datasets

[5] We are using the Alaska Dataset to fine-tune our pre-trained networks. This contains a large number of unaltered images, Cover image, and stegimages generated using using one of three steganography algorithms (DCT, JUNIWARD, UERD). Each embedding algorithm is used with the same probability.

5. Overview of Steganalysis for above steganographic methods

After embedding the image with J-Uniward steganographic method we then tried searching for the best method to do steganaylsis. [17] suggested a CNN that performs best with the J-UNIWARD dataset. It has been demonstrated that deep CNN can beat feature-based methods except in very difficult cases. Due to our computational in capabilities we couldnt implement this algorithm.But as we checked the contest [5] we found out that SRNET in combination with Efficient Nets gives us the best result.The paper for the same [9]. We found out the winning solution got an accuracy of 0.936 with the above setting. And the second winning solution got the accuracy of 0.932 which used efficient net (B7).

5.1. SRNET

[9] SRNet is a deep CNN that can be used for image steganalysis. SRNet is trained on a dataset of high-resolution images and their corresponding steganographic images, learning to distinguish between original and steganographic images based on the subtle differences in image features.

The architecture of SRNet for steganalysis on the J-UNIWARD dataset typically involves the following steps:

1. **Pre-processing:** The input images are pre-processed to ensure that they are in a suitable format for the network. This may include resizing the images to a standard size, and normalizing the pixel values.

2. **Feature extraction:** The pre-processed images are passed through a series of convolutional layers, which extract features from the input images. The features are then pooled and passed to the next layer.

3. **Classification:** The output of the feature extraction layer is fed into a classification layer, which uses a softmax function to classify the images as either stego or cover. The softmax function outputs a probability score for each class, and the class with the highest score is taken as the prediction.

4. **Training:** The network is trained using a large dataset of high-resolution cover images and their corresponding J-UNIWARD stego images. During training, the

network learns to distinguish between stego and cover images based on the differences in the extracted features.

5. **Testing:** The trained network is evaluated on a separate set of high-resolution cover images and their corresponding J-UNIWARD stego images. The performance of the network is measured in terms of accuracy, which is the percentage of correctly classified images.

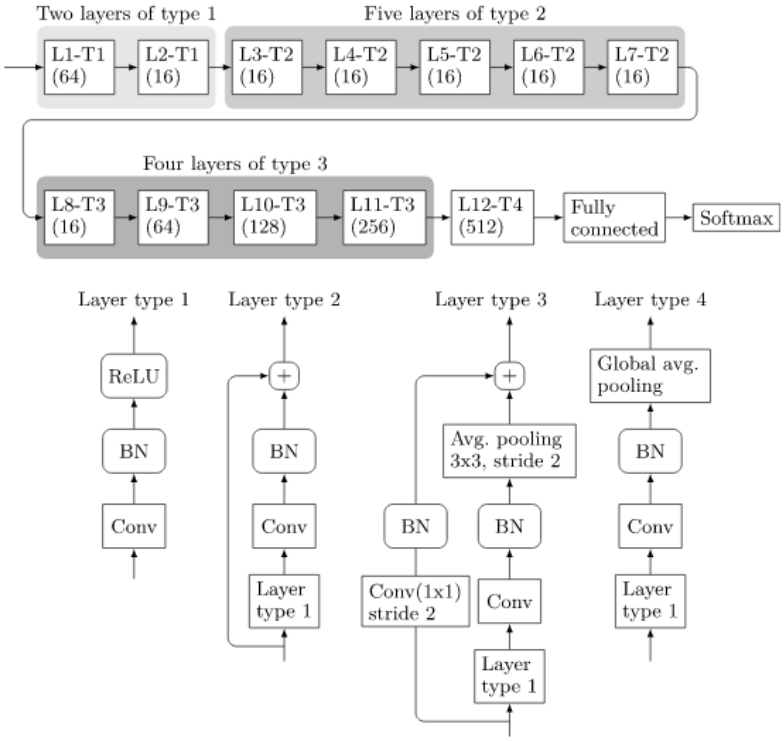

Figure 1. SRNET architechture

The front part of SRNet contains seven residual layers in which pooling has been disabled to allow the network to learn relevant "noise residuals" for different types of embedding changes in both spatial and JPEG domains.SRNet is the first steganalysis network that makes use of the selection channel for JPEG domain steganalysis, a task achieved by adding a bound on L1 embedding distortion to the feature maps outputted by the first layer to reinforce the output of neurons that are most affected by embedding.

5.2. EfficientNet-B7

[15]The architecture of EfficientNet B7 is a scaled-up version of the base EfficientNet architecture. The base architecture consists of a stem convolutional layer, followed by a series of repeated blocks that are designed to increase the depth and width of the network gradually. Each block in the architecture includes a combination of convolutional layers, batch normalization, and activation functions, followed by a pointwise convolutional layer that reduces the number of channels.

EfficientNet B7 has seven blocks, which makes it deeper and wider than the smaller versions in the EfficientNet family. The first block of EfficientNet B7 has 64 channels, while the last block has 640 channels. Additionally, EfficientNet B7 uses an efficient channel attention (ECA) mechanism, which recalibrates channel-wise feature responses. The ECA module is inserted after each convolutional layer, allowing the network to model channel dependencies more efficiently.

The architecture of EfficientNet B7 also includes a global average pooling layer that reduces the spatial dimensions of the output to a single feature vector. This is followed by a fully connected layer and a softmax activation function that produces the final classification scores.

Overall, the EfficientNet B7 architecture is a powerful and efficient neural network architecture that has achieved state-of-the-art performance on a range of computer vision tasks.

5.3. Results

We are using the above-mentioned pre-trained models on the ALASKA dataset [5] for the task of steganalysis.

6. Overview of the application that we are building

We train and test a model for decoding the steganographic message . For that we are using a flask python server we use the pretrained models as the backend model. When a user uploads an image we send an image to the server using a byte encoded format and then decode it in the server. Then we run the steganalysis on the image then our application displays the decoded text image to the user. we have written the code for the website which is stil ongoing.

7. Conclusion

We focused more on the J-UNIWARD steganography as we found out it is more secure method. And we explore some of the datasets such as ALASKA dataset on which the above methods are applied. And we aim to run some of the existing models on the dataset. Further we are looking to do steganaylsis of these images using the deep learning techniques.

We aim to implement the steganalysis technique using the efficient net - B7 and SRNET on the alaska 2 dataset and demostrate it as we found out it will be better. We will be integrating this work in our application that we are building.

8. Final Report

8.1. What We did?

8.1.1 Steganalysis Model

We have trained a model for detecting whether any message was encoded using JUNIWARD in the given jpeg image. We have used a metric as score which describes how likely this image contains hidden data: the higher the score, the more it is assumed that image is encoded.

We have use Efficientnet-b2 pretrained model and fine-tuned on it (Efficientnet-B4 and above versions need higher GPU power necessity, So we cannot train on it) and we trained our model with the data present in the kaggle($>$ 32GB).The steganalysis task is done using binary cross-entropy loss and the AdamW optimizer with 10^4 weight decay, for 60 epochs using a cosine learning rate scheduler with a start LR of 0.001 and end LR of 2×10^5. We used a minimum batch size of 16, which was increased for smaller architectures to speed up training. The JPEG images were decompressed to RGB color space without rounding or clipping. After training, we chose the best checkpoint based on the wAUC metric on the validation set. With full training data our model runs for over 2 days is what we expect to give accuracy. With the references we referenced we expect our model to perform with a score of 0.8.

We implemented SRNet model based on the above architecture(in the 1), and we trained our data on it. But Efficientnet outperforms the SRNet model.

8.1.2 Steganography model

1. Firstly we tried searching for resources to implement the Juniward steganography model. But any where we didnt find any we found matlab codes written by authors but we didnt able to convert it to python code.

2. We have explored many methods other than Juniward but as we found Juniward is one most secure method so we learnt about it and wanted to implement on our own.

3. So at last we found a code [1] from the github. It suggested using DCT, DWT, Daubechies Wavelet functions to embed a message and it has many other methods other than J-uniward. We have took that code and learnt about the code appropriately commented each and every function and written it in a proper way and extracted the part of J-uniward.And there are some errors in the code we hacve also cleared those issues.

4. While doing so we have learnt many things regarding how steganography works and how it is related to concepts that we learn in class.

5. We have rewritten the code in a proper to use do steganography on our image.

6. Note this is not the exact and full fledged implementation of the J-UNIWARD [12]. But an approximate implementation.

8.1.3 Our App

Overview of the app how it works.

1. Our App contains home page which has options of steganography and steganalysis.

2. We can click on the steganography and embed the images. We can get the cover images from alaska2 dataset that is given in the kaggle competition[].

3. And we can just upload the image and encode some data in the images. Using the function we descrbibed above for the Juniward.

4. After Doing we get the stego image back. We can test the same if it was been encoded or not by sending the image to the steganalysis section of the web App.

5. There we can see if our image was encoded with data or not using the score.

How to run the Application

1. Install python version 3.8

```
sudo apt-get install python3.8
```

2. Change the Directory to the root directory of the project and run the following command in the terminal

```
python3 -m venv myenv
```

3. Activate the virtual environment

```
source myenv/bin/activate
```

4. Install all the requirements or the libraries

```
pip3 install -r requirements.txt
```

5. Run the application

```
python app.py or flask run
```

6. Then in browser go to the following link http://localhost:5000/ or whatever port it gives

8.2. Results

8.2.1 Our Model

1. We expected a score of around 0.8 when submitted in kaggle but it gave around 0.6 as we trained our model with less data.

2. We have got the following results for our best trained model:

$$Acc : 0.5978571428571429$$
$$Score : 0.583808712927487$$

8.2.2 Our App

1. When we sent a cover image from the alaksa2 dataset and made a stego image from that.

2. Sent cover image 000001.jpg from alaska dataset to the model for steganalysis returned score of **-0.4**

3. Sent same image for steganography using our JUNIWARD code and then sent it for the steganaylsis it returned a score of **+0.6**

4. Similary we have tested for other images as well they have been so many good results as above.We are able to detect whether a particular image is a stego image or not.

9. Contributions

Note: We have review many research papers and codes on kaggle to understand the steganalysis process.

1. **AI20BTECH11010** Traditional steganographic part of steganography and Steganography Code for JUNIWARD,

2. **AI20BTECH11015** Literature review of DL based steganography and steganalysis and Fine tuning the pretrained model and helped in integrating the trained model to our web app.

3. **AI20BTECH11019** Further Literature review of the DL based steganalysis.Website coding and Integration of the above both(steganography and steganalysis) codes in the website.

4. **EE20BTECH11025** ALaska2 dataset, Kaggle codes exploration. Steganography Code for JUNIWARD.

References

[1] Implmentation of juniward python code. 7

[2] N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. 3, 4

[3] Rita Chhikara Ankita Gupta and Prabha Sharma. A review on deep learning solutions for steganalysis, 2023. 2, 3

[4] H. B. Bahar and Ali Aboutalebi. Image steganography, a new approach for transferring security information, 2008. 2

[5] Dataset for Image Steganalysis. Alaska 2-. 4, 6

[6] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi. Using statistical image model for jpeg steganography: Uniform embedding revisited. *IEEE Transactions on Information Forensics and Security*, 10(12):2669–2680, 2015. 3

[7] V. Holub and J. Fridrich. Random projections of residuals for digital image steganalysis, 2013. IEEE Transactions on information forensics and security. 2

[8] J. Kodovsky and J. Fridrich. Steganalysis of jpeg images using rich models, 2012. Media Watermarking, Security, and Forensics. 2

[9] Member Mehdi Boroumand, Mo Chen and Jessica Fridrich. Deep residual network for steganalysis of digital images, 2017. 4

[10] J. Mielikainen. Lsb matching revisited. *IEEE Signal Processing Letters*, 13(5):285–287, 2006. 3

[11] Deshpande Neeta, Kamalapur Snehal, and Daisy Jacobs. Implementation of lsb steganography and its evaluation for various bits. In *2006 1st International Conference on Digital Information Management*, pages 173–178, 2007. 3

[12] Ante Su, Sai Ma, and Xianfeng Zhao. Fast and secure steganography based on j-uniward. *IEEE Signal Processing Letters*, 27:221–225, 2020. 7

[13] Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. Image steganography: A review of the recent advances. *IEEE Access*, 9:23409–23423, 2021. 2

[14] V. Holub R. Cogranne T. Denemark, V. Sedighi and J. Fridrich. Selection-channel-aware rich model for steganalysis of digital images, 2014. IEEE International Workshop on Information Forensics and Security (WIFS). 2

[15] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 6

[16] S. Tan and B. Li. Stacked convolutional auto-encoders for steganalysis of digital images, 2014. Signal and information processing association annual summit and conference (APSIPA), 2014 AsiaPacific. 2

[17] Guanshuo Xu. Deep convolutional neural network to detect j-uniward, 2017. 3, 4