Project Report

CS512-Computer Vision - Fall 2024

Project by:

Sahil Bhaware A20552865

Aniket Chougule A20552748

Abstract:

The security of image steganography is a vital element in analyzing the quality of steganographic algorithms. Steganography has come a long way in the last few years, cleverly to cope with the challenges thrown by steganalysis techniques. Strengthening the security of image steganography involves techniques able to remain undetected by highly advanced steganalysis algorithms, whose development has also relied on machine learning recently. The conventional embedding method in steganography approaches directly inserts secret messages into the content of the image. As a result, this method can leave over-finesse traces that advanced machine-learning-based steganalysis tools can detect.

To counter these detection capabilities, a new concept called Steganography Without Embedding (SWE) has emerged. SWE aims to circumvent detection by avoiding any modifications to the carrier image’s data, thus reducing detectable traces. This project introduces a novel SWE method that leverages Deep Convolutional Generative Adversarial Networks (DCGANs). In this approach, secret information is mapped to a noise vector, and a trained generator neural network creates the carrier image from this noise vector, eliminating the need for embedding operations. Another neural network, termed the extractor, is then trained to successfully retrieve the encoded information from the generated image.

This method demonstrates high accuracy in information extraction while offering a robust defense against detection by state-of-the-art image steganalysis algorithms. The experimental results underline the potential of this GAN-based approach for secure and undetectable image steganography.

Problem Statement:

Traditional methods of image steganography hides secret information in images acting as carriers, which are left with subtle traces. Machine learning-based steganalysis programs are developed which can tell the differences between original and modified images quite easily. This vulnerability makes the secret communication both insecure and detectable. To solve this, the project proposes a method named Steganography Without Embedding (SWE) using Deep Convolutional Generative Adversarial Networks (DCGANs) to create carrier images from scratch without any detectable traces and still allows secure and good information retrieval.

Need for GAN-based Steganography

A. EMBEDDING-BASED STEGANOGRAPHY

- **Least-Significant-Bit (LSB) Matching**: LSB matching hides secret information by replacing least significant bits of each pixel, altering the image's data while preserving its appearance. Machine learning-based steganalysis detects these subtle changes, making LSB matching vulnerable. GAN-based steganography generates new images containing the information without bit-level modification, enhancing resistance to detection.
- **HUGO (Highly Undetectable Stego)**: HUGO embeds data while minimizing distortion by carefully choosing adjustments. Despite precautions, advanced steganalysis tools detect a statistical "fingerprint." GANs generate images with hidden information without modifying existing pixels, eliminating the fingerprint.
- **Wavelet Obtained Weights (WOW)** and **S-UNIWARD**: Both WOW and S-UNIWARD embed data in complex or noisy image regions to hide changes. However, as steganalysis methods improve, they can detect these "hotspots" where data is hidden. A GAN-based approach creates images with hidden information without embedding data in detectable regions.

B. DEVELOPMENT OF STEGANALYSIS

Steganalysis algorithms detect hidden messages in images. Old techniques included LSB embedding in digital photos and Markov-based models for JPEG images. Pevny and Fridrich presented combined features for JPEG steganalysis, including the SPAM model that detected pixel differences using Markov chains.

Deep learning advancements led to more intricate steganalysis models. Qian's CNN-based model and Zeng's hybrid framework contrasted with classic embedding-based steganography, making data hiding more challenging.

C. IMAGE STEGANOGRAPHY WITHOUT EMBEDDING

Steganography Without Embedding (SWE) is a novel concept for the avoidance of detection by digital forensics based on establishing the relationship between cover data and secret data without any physical modifications of the cover data. There are two main SWE methods: cover-selection and cover-synthesis.

- **Cover-Selection Approach:** This method involves selecting natural images from a database that correlate with the secret information, often using hash values to index and retrieve suitable images. While effective, it requires a large image repository and has limited applicability due to the specificity of matching criteria.
- **Cover-Synthesis Approach:** This technique generates new images based on the hidden information, such as using texture synthesis to create images that embed messages. Although it offers innovative ways to conceal data, reliance on specific image types (e.g., textures) may arouse suspicion if overused.

We have used cover-synthesis approach in this project.

D. GANs FOR IMAGE STEGANOGRAPHY

GANs produce new images without the need of pixel modifications, which in turn makes the data hiding secure. With steganography based on GAN, the hidden data becomes imperceptible to the traditional embedding methods, making it difficult for the detection tools of steganalysis. The earlier methodologies such as Hayes, Volkhonskiy, or Tang, relied on embedding methods and were therefore either vulnerable or less efficient. This paper proposes a new method of data hiding that does not include embedding and it is possible thanks to DCGANs which can generate a carrier image from scratch and thus, do not contribute to the detection of the traces.

- **Higher Security:** The generated images consist of concealed information that ae invisible to the eye.
- **Flexibility:** The produced by GANs images have a natural look and can be used in a variety of applications.
- **Resistance to Advanced Detection:** GANs generate images that do not embed data which will make them less detectable by traditional and machine-learning-based steganalysis tools.

Paper Selected:

“A Novel Image Steganography Method via Deep Convolutional Generative Adversarial Networks” by DONGHUI HU, LIANG WANG, WENJIE JIANG, SHULI ZHENG, AND BIN LI published in 2018 by IEEE

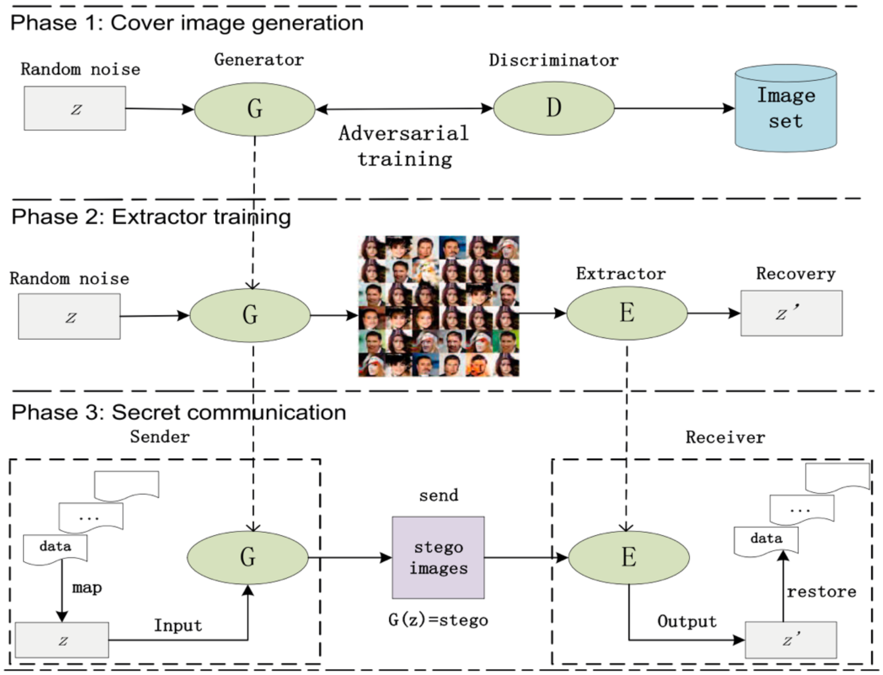
Dataset Used:

CelebA is a large-scale facial dataset that is widely used in computer vision and machine learning. It consists of more than 200,000 images of celebrities with rich annotations, such as the 40 attribute labels (e.g. age, gender, facial features) and 10,000 distinct identities. The dataset showcases a variety of poses, expressions, and backgrounds, making it suitable for tasks such as face recognition, attribute prediction, and generative modeling. Its diversity and labeled attributes render it a particularly valuable dataset for the training of models and their evaluation in tasks that involve facial image processing, including GAN-based image generation and transformation.

Preprocessing:

- **Image Resizing :** The images were resized to 32x32 pixels with 3 color channels (RGB).
- **Data reduced:** Original Dataset contain 200k+,reduced to 3200 images.
- **Normalization:** Adjusted pixel values to the range of values from -1 to 1 to improve training stability of the model.
- **Data Shuffling:** Used a shuffle buffer size of 1,000 to randomize the order of pictures during training, and that is over than the order of images during training while letting learning be as general as possible.

Implementation in the paper



Our Implementation

PHASE 1: PREPROCESSINGAND DATASET PREPARATION

This part deals with the picking of a suitable dataset that might be, for instance, CelebA, resizing images to the same (64x64) size, and normalizing them for training. Besides, it also treats the establishment of an efficient data pipeline to load and augment the images.

PHASE 2: BUILDING THE DCGAN MODELS

During, this phase building of the core components of GAN ; the generator model generating images by using noise as input, and deconvolution layers and activations are used. Discriminator model is that model which classifies the images as either real or fake, by using convolutional layers and binary classification. They both, however, are compiled and prepared for adversarial training.

PHASE 3: TRAINING THE DCGAN

This part involves the generator and discriminator in an adversarial loop being between them. The discriminator separates real from generated images, the generator makes the discriminator confused. The generator that has been trained now can produce stego images.

PHASE 4: IMPLEMENTING THE EXTRACTOR MODEL

This is the place when an extractor model is written and then taught to decode hidden message. It involves the usage of convolutional layers to reverse the process of steganography which leads to retrieving the encoded noise vectors from the stego images generated by the GAN.

PHASE 5: SECRET MESSAGE HIDING (SENDER SIDE)

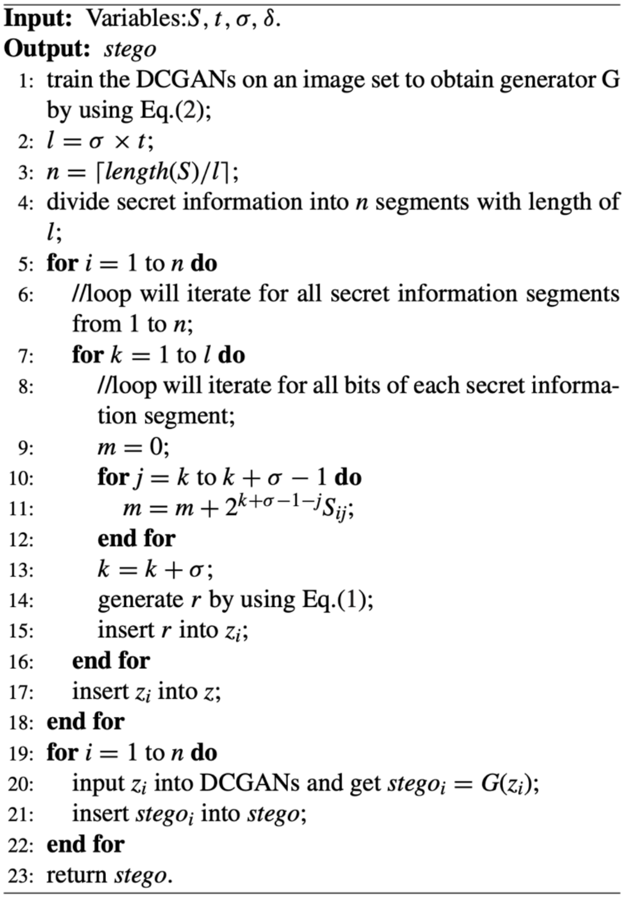
The sender in this phase is the one who applies the trained generator sending a binary message into images by mapping the message to noise vectors and producing stego images. These images which are now with the hidden data can be either stored or sent to the receiver.

PHASE 6: SECRET MESSAGE EXTRACTION (RECEIVER SIDE)

The last phase is about decoding the hidden message embedded in the stego images. The receiver then employs the extractor model to get the noise vectors from the images and finally, from these vectors, he/she can create the original binary message, which is done in a secure and accurate way.

Algorithms

1. Algorithm to generate stego images



```
**Input:** Variables:$S, t, \sigma, \delta$.
**Output:** *stego*
train the DCGANs on an image set to obtain generator G by using Eq.(2);
$l = \sigma \times t$;
$n = \lceil length(S)/l \rceil$;
divide secret information into $n$ segments with length of $l$;
for $i = 1$ to $n$ do
    //loop will iterate for all secret information segments from 1 to $n$;
    for $k = 1$ to $l$ do
        //loop will iterate for all bits of each secret information segment;
        $m = 0$;
        for $j = k$ to $k + \sigma - 1$ do
            $m = m + 2^{k+\sigma-1-j} S_{ij}$;
        end for
        $k = k + \sigma$;
        generate $r$ by using Eq.(1);
        insert $r$ into $z_i$;
    end for
    insert $z_i$ into $z$;
end for
for $i = 1$ to $n$ do
    input $z_i$ into DCGANs and get $stego_i = G(z_i)$;
    insert $stego_i$ into $stego$;
end for
return $stego$.
```

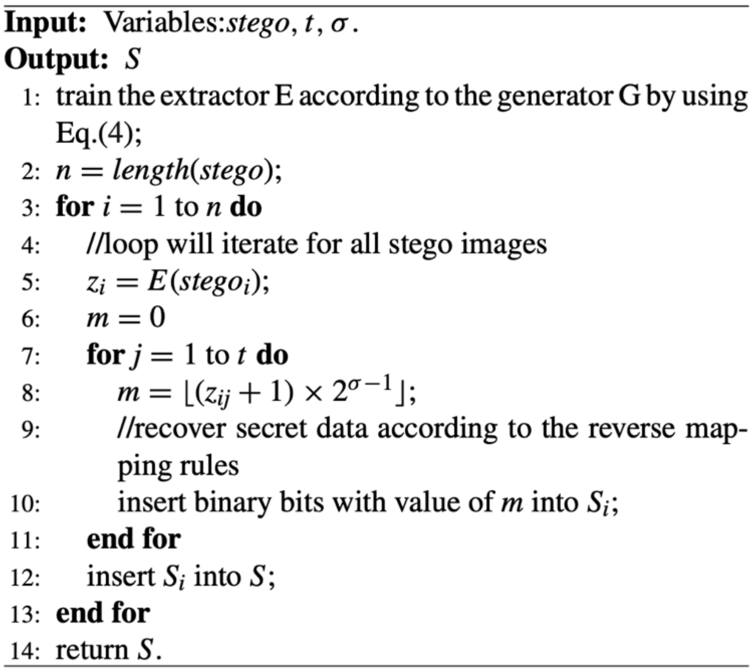
Algorithm Strengths

1. **High Security:** Generates images from scratch, minimizing detectable traces.
2. **Scalable:** Handles large data efficiently through segmentation.
3. **Realistic Outputs:** Uses DCGANs for high-quality, believable images.

Algorithm Weaknesses

1. **Computationally Intensive:** Training DCGANs requires significant resources and time.
2. **Noise Sensitivity:** Encoding into noise vectors demands precise tuning to ensure quality and capacity.

2. Algorithm for data Extraction and Recovery



```
Input: Variables: $stego, t, \sigma$.
Output: $S$
train the extractor E according to the generator G by using Eq.(4);
$n = length(stego)$;
for $i = 1$ to $n$ do
    //loop will iterate for all stego images
    $z_i = E(stego_i)$;
    $m = 0$
    for $j = 1$ to $t$ do
        $m = \lfloor (z_{ij} + 1) \times 2^{\sigma - 1} \rfloor$;
        //recover secret data according to the reverse mapping rules
        insert binary bits with value of $m$ into $S_i$;
    end for
    insert $S_i$ into $S$;
end for
return $S$.
```

Algorithm Strengths

1. **Accurate Recovery:** Effectively retrieves secret data with minimal errors through trained extractor and reverse mapping.
2. **Seamless Integration:** Works efficiently with DCGAN-generated stego images for consistent performance.

Algorithm Weaknesses

1. **Dependent on Stego Image Quality:** Poor-quality stego images can hinder accurate data recovery.
2. **Computational Overhead:** Training the extractor adds complexity and requires additional resources.

Key Components:

Generator:

The generator network takes random noise (z) as input and produces an image. It then passes this image to the discriminator network (D), which classifies images as real or fake. Adversarial training helps the generator generate images that the discriminator can't distinguish from real ones, improving image quality and realism. This phase ensures the generator's output is high quality, which is crucial for carrier generation in hidden communication. information.

Discriminator:

This neural network generates data samples like images from random noise inputs. Its aim is to generate outputs which are indistinguishable from real data, therefore, "fooling" the discriminator and getting it to classify them as if they were real.

Extractor:

The extractor, trained to encode the original noise vector (z') from generated images, can still discern hidden messages despite the generation process being perfectly reverse learned. This step establishes a framework for reliably extracting hidden messages. The generator, acting as a decoder, ensures safe and consistent data retrieval of the hidden information.

Modifications:

Use of Brown Corpus: Earlier, we used randomly generated noise during training; instead, we now utilize sentences from the Brown corpus. These sentences are converted into binary format to create noise vectors, introducing a more structured and meaningful representation of data, thereby improving the model's generalization and robustness in extracting messages.

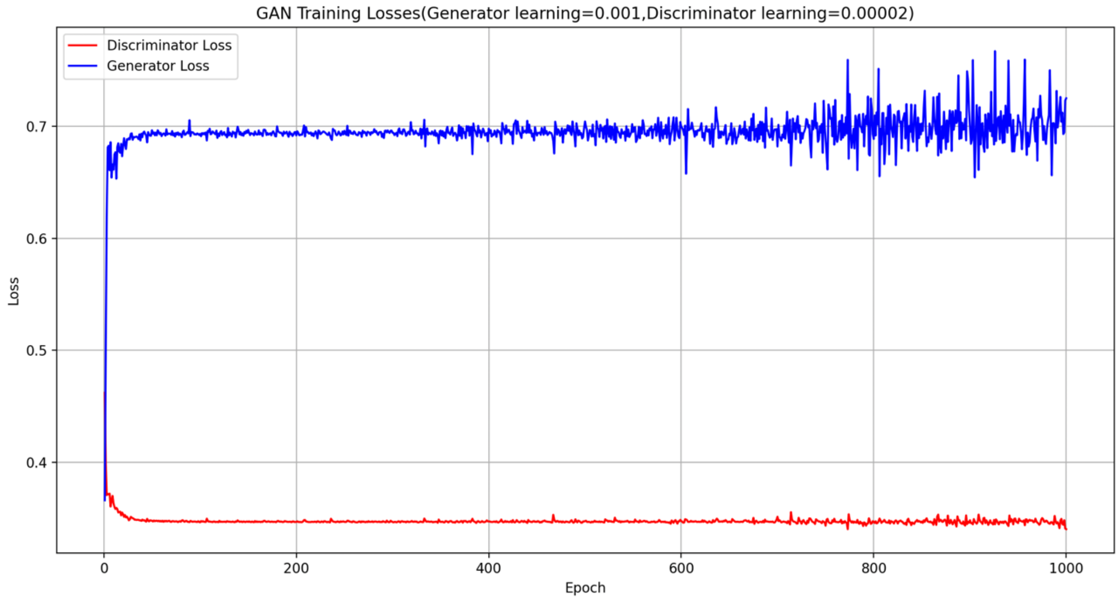
Handling Zero Padding in Noise Integration: To address the problem of zero padding while integrating the message into the noise, we repeated the noise vector to match the required noise dimension of 100. This ensured that the input dimensions remained consistent, preventing training instability and improving the quality of the generated carrier images.

Process:

- **Sender:** The sender encodes the secret message through mapping it to a noise vector (z). The noise vector is the one used to describe the mechanisms in the generator (G) whose value is originally randomized based on the runtime conditions. However, a perceived normal image is produced playing the role of "stealthy," which has a secret message discrepancy instead of being a real image.
- **Transmission:** The generated stego image is shared from the sender to the receiver through the channel. As the image comes out naturally, this technique avoids attack and gives no notice to the operator.
- **Receiver:** The operator harnessing the decoding algorithm emanating from the extractor to reconstruct the vector of the noise (z') initiates the process. The gleaned vector can then be deciphered back into the original secret data. thus, the information has been successfully transmitted securely.

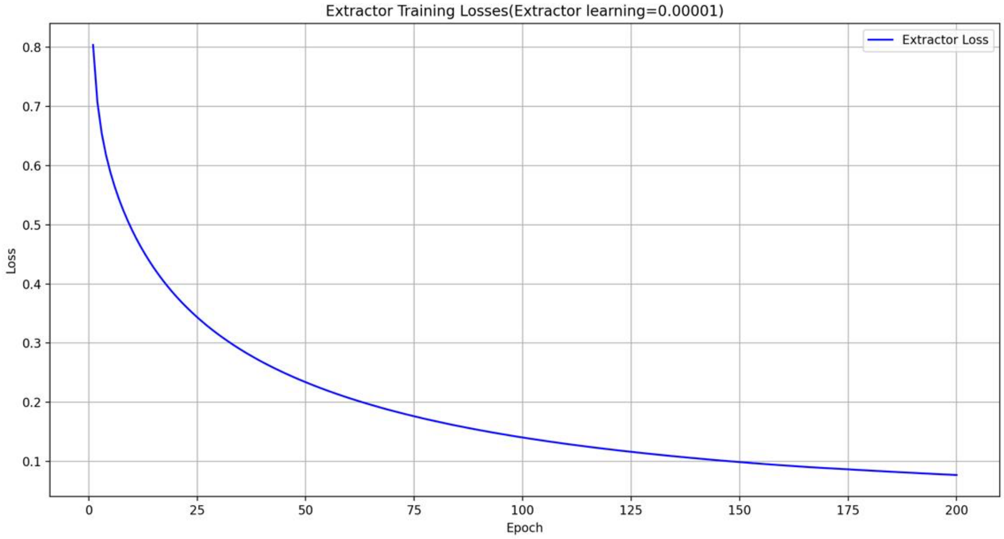
Observations

1.GAN Training losses



- The discriminator loss decreases rapidly and stabilizes at a low value, indicating effective learning to distinguish real from generated data.
- The generator loss stabilizes after an initial increase, with fluctuations suggesting a balanced adversarial training process.
- The difference in learning rates (generator: 0.001, discriminator: 0.00002) ensures stability and controlled training.
- Training reaches equilibrium around 200 epochs, with both models maintaining a competitive balance. Summarizing, the losses indicate successful GAN training without divergence or collapse.

2.Extractor Training Losses



- The extractor loss starts high (around 0.8) and steadily decreases throughout the training process, indicating continuous improvement in the performance of extractor.
- The loss exhibits a smooth, exponential decay pattern, which suggests effective learning and convergence without instability or oscillations.
- By the 200th epoch, the extractor loss drops to (around 0.1), indicating high accuracy in extracting information from generated images.
- The chosen learning rate of 0.00001 appears appropriate, balancing learning speed and stability.
- The curve indicates that the extractor model successfully converges and achieves minimal error.

Challenges Faced

1. **Resource Limitations:**
   Without access to TPUs and limited computational power on our laptops, we trained on GPUs which consequently resulted in longer training and slower experimentation cycles.
2. **Lack of Prior GAN Experience:**
   Our team's previous knowledge of GANs was nonexistent, thus we were facing ignescent problems that absorbed a lot of time for experiments and testing different strategies which delayed the project remarkably.
3. **Handling Large Dataset:**
   The CelebA dataset recommended in the research paper was extensive and feature-rich, requiring significant training time and a higher number of epochs. To overcome this, we preprocessed and reduced the dataset, enabling us to complete the project within the constraints of our available resources.

Conclusion

The SWE method via Deep Convolutional Generative Adversarial Networks (DCGANs) for secure and undetectable image steganography is a new way of doing that. With the SWE approach, the embedding operation is completely eliminated and we can generate images from scratch by using a technique called generative modeling, which makes it impregnable to advanced steganalysis techniques.

Some key modifications include using the Brown Corpus to form structured noise vectors and zero-padding issues are solved by repeating the noise to make the size requirements last, which greatly increases the model’s robustness and the ability to generalize. The experimental results included good GAN training, high-quality image generation, and error-free information extraction.

With the advantages of higher security, flexibility, and resistance to detection, this method takes a lead in embedding-free steganography. It could further its effectiveness by improving its computational efficiency, which would be suitable for more secure communication situations.