Deep Steganography: Hiding Images within Images

03CS6902 Mini Project
Design Report

CHN20CSIP07 Sree Lekshmi B S
sreelekshmibs16@gmail.com
M. Tech. Computer Science (Image Processing)

Department of Computer Engineering
College of Engineering Chengannur
Alappuzha 689121
Phone: +91.479.2165706
http://www.ceconline.edu
hod.cse@ceconline.edu

Abstract

This project aims to present a system to hide a full color image inside another of the same size with minimal quality loss to either image. For that we design deep neural networks which are simultaneously trained to create the hiding and revealing processes. The full system is a series of three networks that are trained as a single large network. The system is trained on images using ImageNet database and works well on natural images from a wide variety of sources. The challenge of good information hiding arises because embedding a message can alter the appearance and underlying statistics of the carrier. This work also attempt to maintain quality of images. With this work, not only the hidden information be kept secure, but the system can be used to hide even more than a single image. Unlike many popular steganographic methods that encode the secret message within the least significant bits of the carrier image, our approach compresses and distributes the secret image's representation across all of the available bits.

Contents

Chapter 1

Introduction

Information hiding is a technique of hiding secret data using redundant cover data such as images, audios, movies, documents etc. Information hiding is most commonly associated with secretly planning and coordinating criminal activities through hidden messages in images posted on public sites. Beyond the multitude of misuses, hiding information can be used for practical positive applications. For example, hidden images used as watermarks embed authorship and copyright information without visually distorting the image. Cryptography and steganography are main methods used to hide or protect secret data. However, they differ in the respect that cryptography makes the data unreadable or hides the meaning of the data while steganography hides the existence of the data. Steganography often use cover images to hide data.

1.1 Proposed Project

This project presents an efficient system to hide a full color image within another of same size. This work aims to hide an image without altering the appearance of cover image. It is implemented by deep neural networks which are simultaneously trained to create the hiding and revealing processes and are designed to specifically work as a pair.

1.1.1 Problem Statement

This project aims to hide large amount of information within a cover image without losing the quality of both. Also the amount of information hidden will not alter the appearance and underlying statistics of cover image.

1.1.2 Proposed Solution

For effective and efficient hiding of hidden image's information into host image it employs a series of three deep neural networks namely Preparation network, Hiding Network and Revealing Network. These network determines where to place the hidden information as well as how to compress and represent it. The hidden image is dispersed throughout the bits in surrounding pixels and across all the color channels. A decoder network, that has been simultaneously trained with the encoder is used to reveal the hidden image. The Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM), a perceptual metric are use to quantify image quality degradation between the original and reconstructed images.

Chapter 2

Project Design

The goal of this project is to visually hide a full N × N RGB pixel secret image in another N ×N RGB cover image with minimal distortion to the cover image. Though steganography is often conflated with cryptography, in our approach the closest analogue is image compression through auto-encoding networks. The trained system must learn to compress the information from the secret image into the least noticeable portions of the cover image[1]. The architecture of the proposed system is shown in Figure 2.1. The three components in the systems are Preparation Network, Hiding Network and Revealing network and are trained as a single network.

Preparation-Network

The first component is the Preparation-Network that prepares the image to be hidden. The main function of this network is to transform the RGB-pixels of the hidden image into features that can be used by the Hiding-Network.

Hiding Network

The second and main component is the Hiding Network. The Hiding-Network receives the output of the Preparation-Network and the host image as input. The input is formatted as an$N \times N$

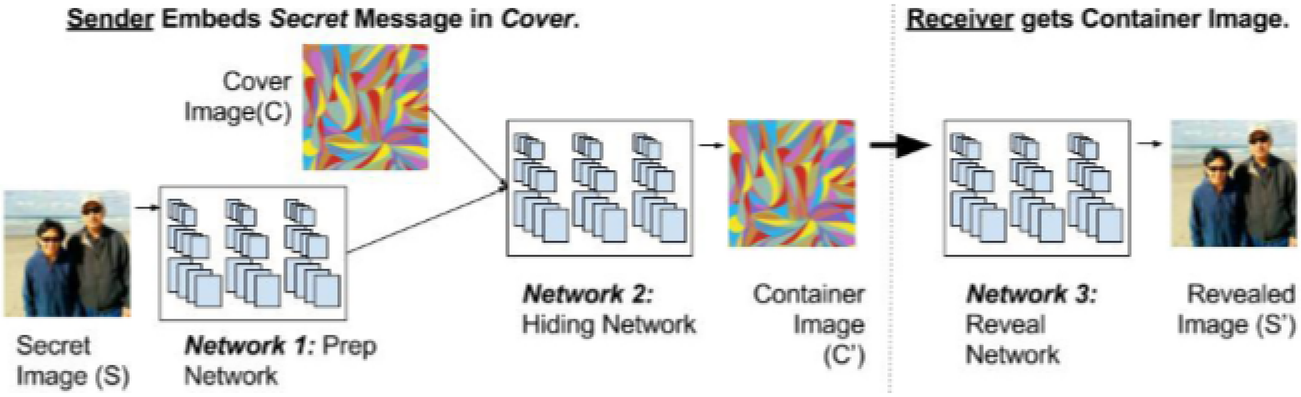
Sender Embeds Secret Message in Cover.
Receiver gets Container Image.
Cover Image(C)
Secret Image (S)
Network 1: Prep Network
Network 2: Hiding Network
Container Image (C')
Network 3: Reveal Network
Revealed Image (S')

Figure 2.1: Architecture of proposed system

pixel field with depth concatenated RGB channels of the host image and the transformed channels of the hidden image. The output of this network is the Container image ($N \times N$, RGB pixels). The container image should appear as similar to the host as possible, while also containing enough information to recreate the hidden image.

Revealing Network

The third component is the Reveal-Network that is responsible for extracting the hidden image from the container. Though this network is used only by the receiver all three components are trained as a single network.

The system is trained by reducing the error shown below (H and S are the cover and secret images respectively, and is how to weigh their reconstruction errors):

$$\epsilon(H, H', S, S') = ||H - H'|| + \beta||S - S'||$$

By propagating this error signal to both the Preparation and Hiding networks the representations formed early in the system encode information about the hidden image.

Our aim is to encode a large amount of information into limited visually noticeable artifacts. The images used in the study are composed at each pixel of 24 bits (8 × (R,G,B)). We flip the first bit of the R channel of all the pixels in the container image, we can measure its effects on the reconstructions on the container image itself and also by propagating the modified image through reveal network on the reconstruction of the secret image. We can see that a bit flip in any bit position in any color channel of the container image has an effect across all color channels in the hidden image's reconstruction[2]. The information for the hidden image is spread across the color channels, the reason it was not detected by simply looking at the LSB. In addition to distributing the hidden image information across the color-bits the information is also spread in the spatial dimension.

So the representation for the hidden image is distributed both in surrounding pixels and in color bits. The encoding for each pixel of the hidden image is distributed in pixels that are up to a distance of 7 away from the corresponding pixel in the container image. Second, the amount of spatial distribution is directly related to the neural network architecture and the size of the convolutions.

To ensure that the networks do not simply encode the secret image in the LSBs, a small amount of noise is added to the output of the second network during training. The noise was designed such that the LSB was occasionally flipped; this ensured that the LSB was not the sole container of the secret image's reconstruction.

2.1 Hardware & Software Requirements

Operating System	:	Any Operating System
Supporting software	:	Python
Processor	:	Intel Core i5 11th Gen 4.50GHz
RAM	:	8GB
Monitor	:	Any colour monitor

Chapter 3

Project Progress

Below are the work done so far:

1. Studied the reference paper.
2. Conduct literature survey of related works.
3. Installed Python and started learning it.
4. Collect dataset.
5. Made the design of the project.

3.1 Work Schedule

Below are shedule of work (till August 10)

1. Identify suitable project area and topic.
2. Studied the reference paper well.
3. Obtained IC and choose guide.
4. Read some related papers of the topic.
5. Analysed various methods that can be used in this project.
6. Installed Python and started learning it.
7. Make design of the project.

Work sheduled for coming time period.

1. Start implementation.
2. Implement each module of the project.

References

[1] Shumeet Baluja: Hiding Images within Images, IEEE Trans. Pattern Anal. Mach. Intell, 2020

[2] A.K Jain and U. Uludag: Hiding Biometric Data, IEEE Trans. Pattern Anal. Mach. Intell, 2003

[3] S. Baluja: Hiding images in plain sight: Deep steganography, Proc. Neural Inf. Process. Syst, 2017