Digital Image Processing(UE18EC317)

Project Title: **Image Steganography**

Parth Bansal (PES1201801325)
Akash G (PES1201801970)
Shravan Kumar V.R (PES1201801076)

Table of Contents:

(Click to go to Respective Sections)

1. Abstract

The project deals with learning about the concepts of various types of steganography available. Image steganography is performed for images and the concerning data is also decrypted to retrieve the message image. Since this can be done in several ways, Image Steganography is studied and one of the methods is used to demonstrate it.

Image steganography refers to hiding information i.e. text, images, or audio files in another image or video files. This project report intends to give an overview of image steganography, its uses, and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects on which steganographic techniques are more suitable for which applications.

2. Acknowledgement

We would like to thank Prof. Lavanya Krishna and Prof Shruthi M.L.J for their guidance and assistance throughout the course of digital image processing. This project was motivated only because of the learnings of this course and we appreciate all the support we have received in this time.

It is to be noted that any errors in this report, and/or in the project are the authors' only, and must not be associated to these distinguished personalities.

3. Introduction

Steganography is the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message. The word "Steganography" is of Greek origin and means "covered or hidden writing". An encrypted

file may still hide information using steganography, so even if the encrypted file is deciphered, the hidden message is not seen. The advantage of Steganography over Cryptography alone is that messages do not attract attention to themselves, to messengers, or the recipients. A steganographic message is often first encrypted by some traditional means and then the hidden text is modified in some way to contain the encrypted message, resulting in stegotext.

Steganography is of different types:

1. Image Steganography
2. Video Steganography
3. Audio Steganography
4. Text Steganography
5. Email Steganography
6. Network Steganography

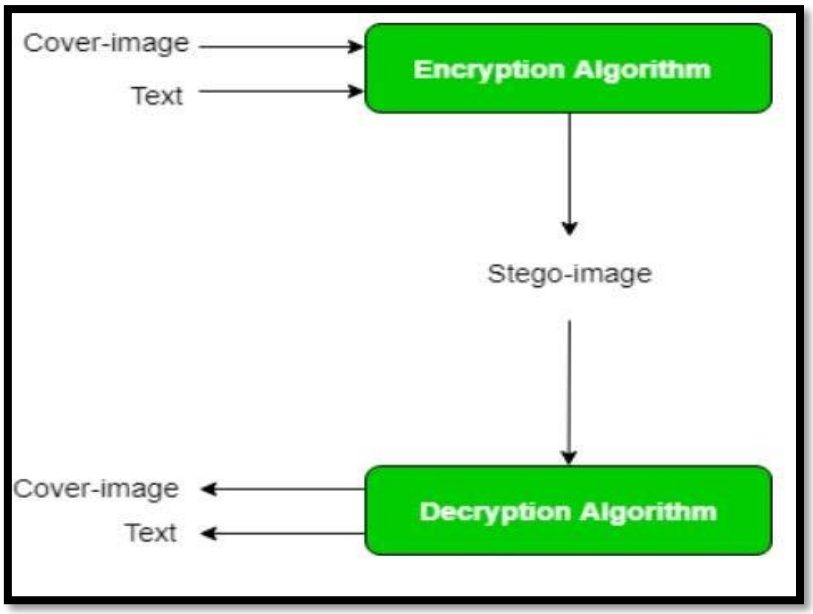
In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used.

3.1 Image Steganography:

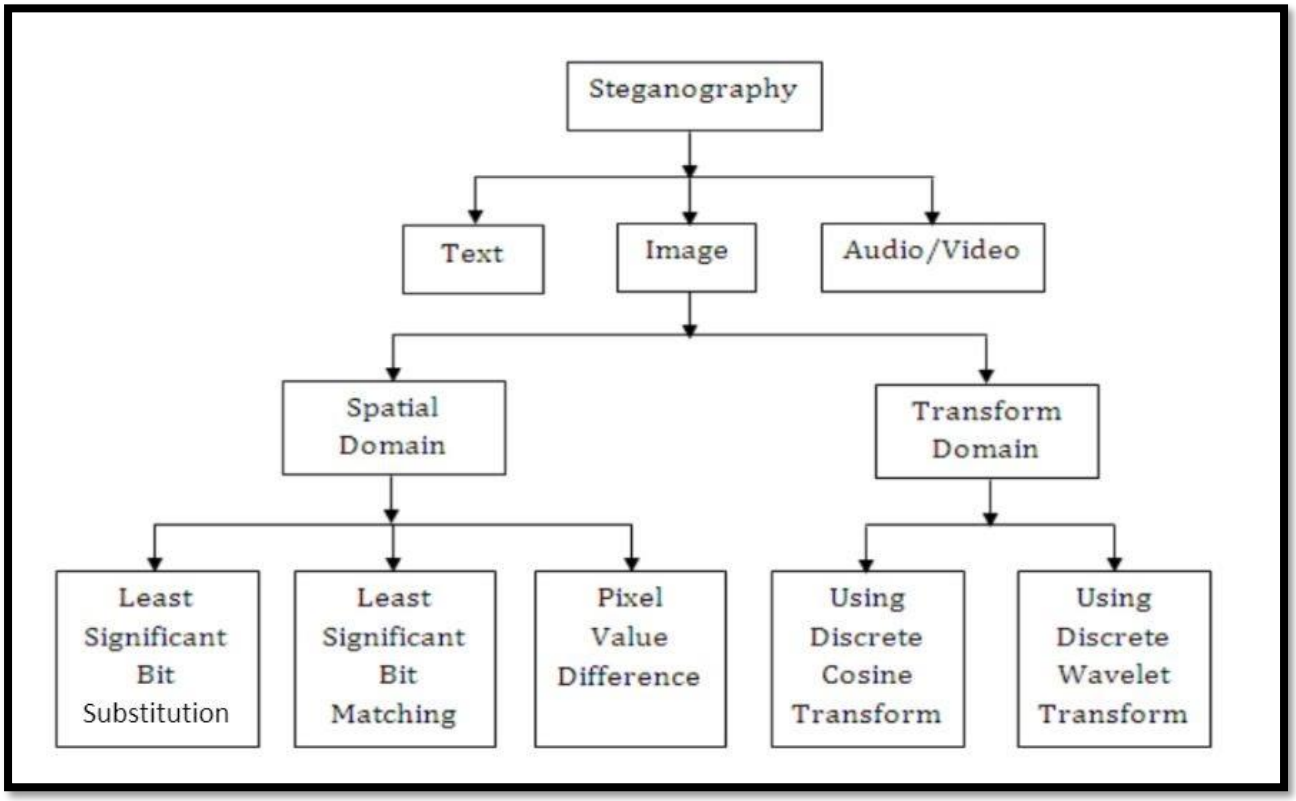
As the name suggests, Image Steganography refers to the process of hiding data within an image file. The image selected for this purpose is called the cover-image and the image obtained after steganography is called the stego-image.

3.2 How Image Steganography can be achieved?

An image is represented as an N*M (in case of grayscale images) or N*M*3 (in case of color images) matrix in memory, with each entry representing the intensity value of a pixel. In image steganography, a message is embedded into an image by altering the values of some pixels, which are chosen by an encryption algorithm. The recipient of the image must be aware of the same algorithm to know which pixels he or she must select to extract the message.



Detection of the message within the cover-image is done by the process of **steganalysis.** This can be done through comparison with the cover image, histogram plotting, or noise detection. While efforts are being invested in developing new algorithms with a greater degree of immunity against such attacks, efforts are also being devoted towards improving existing algorithms for steganalysis, to detect the exchange of secret information between terrorists or criminal elements.



3.3 Different Methods for Image Steganography:

1. **Spatial Domain:** It includes LSB (Least Significant Bit) Steganography. The spatial methods are most frequently employed because of fine concealment, the great capability of hidden information, and easy realization. LSB Steganography includes two schemes: Sequential Embedding and Scattered Embedding.

2. **Transform Domain:** The method of transform domain Steganography is to embed secret data in the transform coefficients.

3.4 Least Significant Bit Matching Method:

The least significant bit (LSB) method is a common, simple approach to embedding information in a cover file.
In steganography, the LSB matching method is used since every image has three components (RGB). This pixel information is stored in an encoded format in one byte. The first bits containing this information for every pixel can be modified to store the hidden text. For this, the preliminary condition is that the text to be stored has to be smaller or of equal size to the image used to hide the text. LSB based method is a spatial domain method. But this is vulnerable to cropping and noise. In this method, the MSB (most significant bits) of the message image to be hidden are stored in the LSB (least significant bits) of the image used as the cover image. It is known that the pixels in an image are stored in the form of bits. In a grayscale image, the intensity of each pixel is stored in 8 bits (1byte). Similarly, for a color (RGB-red, green, blue) image, each pixel requires 24 bits (8 bits for each layer).
The Human visual system (HVS) cannot detect changes in the color or intensity of a pixel when the LSB bit is modified. This is psycho-visual redundancy since this can be used as an advantage to store information in these bits and yet notice no major difference in the image.

4. Review of Literature:

Steganography is used to achieve one of the following goals:

•Concealing information [3];

•Hiding any trace of information being transmitted.

This keeps any third-party observers unaware of its presence. In both cases, choosing the most suitable data carrier is crucial for secrecy. The two most important properties for a carrier/vector are:

• Popularity: the used vector should be not considered as an anomaly itself, potentially unmasking the existence of the hidden communication;

• Conspicuousness: A third-party unaware of steganographic procedures should not be able to see the modifications in the vector.

Although steganography applies to all data objects that contain redundancy, we consider images. People transmit digital pictures over email and other Internet communication very frequently. Moreover, steganographic systems for the images are not affected by visual attacks(Visual attacks mean that you can see steganographic messages because they overwrite visual structures) Neil F. Johnson and Sushil Jajodia [4], for example, showed that steganographic systems for palette-based images leave easily detected distortions.

Based on the aforementioned points, we have chosen the Least Significant Bit (LSB) method for our project, since the sensitive data is concealed extremely well[5], and the popularity of images as a method of communication allows for reduced suspicion of carrying any concealed data. In addition to that, we are using PNG images, because of their lossless nature[7].

5. Implementation

5.1) Code:

```
%Encoding the message
Input = imread('TestImage.jpg');
Cover = rgb2gray(Input);
[row,column] = size(Cover);
L = 256;
Stego = Cover;
Message = input('Enter the message to be hidden: ','s');
len = strlength(Message)*8;  %Each character will take 8 bits so total number of bits in the message will be len
Ascii = uint8(Message);   %ascii is a vector having the ascii value of each character
Binary_separate = dec2bin(Ascii,8);   %binary_separate is an array having the decimal representation of each ascii value
Binary_all =”’;  %binary_all will have the entire sequence of bits of the message
for i=1:strlength(Message)
    binary_all=append(binary_all,binary_separate(i,:));
end
count=1;    %initializing count with 1
for i=1:row
    for j=1:column
        %for every character in the message

        if count<=len
            %Obtain the LSB of the grey level of the pixel          LSB=mod(Cover(i,j),2);
            %Convert the bit from the message to numeric form
            a=str2double(binary_all(count));
            %Perform XOR operation between the bit and the LSB
            temp=double(xor(LSB,a));
            %Change the bit of the stego image accordingly
            Stego(i,j)=Cover(i,j)+temp;
```

```
            count=count+1;
        end
    end
end
subplot(1,2,1);
imshow(Cover);
title('Cover Image');
subplot(1,2,2);
imshow(Stego);
title('Stego Image');

%Decoding the message
count=1;
message_in_bits='';
for i=1:row
    for j=1:column
        %For all the characters in the message
        if count<=len
            %Retrieve the LSB of the intensity level of the pixel
            LSB=mod(Stego(i,j),2);
            %Append into message_in_bits to get bit sequence of message
            message_in_bits=append(message_in_bits,num2str(LSB));
            count=count+1;
        end
    end
end

%Converting the bit sequence into the original message
i=1;
original_message='';
while i<=len
    %Take a set of 8 bits at a time
    %Convert the set of bits to a decimal number
    %Convert the decimal number which is the ascii value to its corresponding
    character
    %Append the obtained character into the resultant string
    original_message=append(original_message,char(bin2dec(message_in_bits(
    1,i:i+7))));
    i=i+8;
end
disp(['The original message is: ',original_message]);
```

5.2) Code Explanation:

5.2.1) Encoding the Image:

The algorithm used to encode the image:

1. Convert the image to grayscale
2. Convert the message to its binary format
3. Initialize output image the same as the input image
4. Traverse through each pixel of the image and do the following:
    - Convert the pixel value to binary
    - Get the next bit of the message to be embedded
    - Create a variable temp
    - If the message bit and the LSB of the pixel are the same, set temp = 0
    - If the message bit and the LSB of the pixel are different, set temp = 1
    - This setting of temp can be done by taking XOR of message bit and the LSB of the pixel
    - Update the pixel of the output image to input image pixel value + temp
5. Keep updating the output image till all the bits in the message are embedded
6. Finally, write the input as well as the output image to the local system.

Firstly, we are converting the input image into a grayscale image, since we are going to be performing image steganography using the LSB (least significant bit) method.
We are then requesting user input for the message to be hidden into the image, and generating the number of bits of the message (each character takes 8 bits).
A sequence of binary digits is generated, with each set of 8 binary values representing the ASCII value of the given character.
For example, if the input message was ‘Hi’, the sequence would be as follows:
ASCII value of ‘H’-72
72 in binary- 01001000
ASCII value of ‘i’-105
105 in binary- 1001000
Hence, the sequence of binary digits is - 010010001001000

To obtain the LSB of every gray level in the image, we are using the mod (remainder) function. If the value of the (number)%2=0, then the number is even and hence has 0 as its least significant bit. Conversely, if the number is odd, it will have 1 as its least significant bit.

To change the least significant bits of the gray levels, we are using Parity as the concept. For this, we are making use of the XOR function. If the LSB of the gray level of the image is the same as that of the bit to be encoded, no change is made (XOR function returns 0 if both the inputs are the same). If the LSB of the gray level is not the same as the bit to be encoded, 0 is replaced by 1, and 1 is replaced by 0 (XOR function returns 1 if both inputs are different).
In this way, the loop takes only the number of pixels that are equal to the length of the input (number of characters*8) and replaces the bits according to the parity. The output image, with the message encrypted, is then displayed.

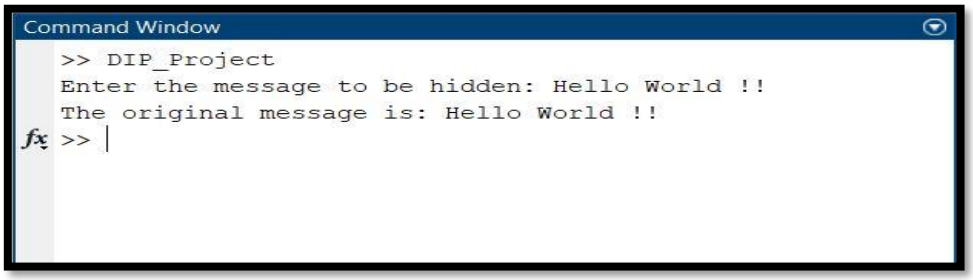
5.2.2)Decoding the Image:

The decoding process is almost identical to the encoding process but in the reversed order.

For all the pixel values with the message encrypted, the least significant bit of each of the intensity values is retrieved. These bits are then combined, and for each set of 8 bits, it is converted into its respective ASCII value to retrieve the character. These characters are then put together into a character array and displayed as the decoded output. In this way, we can retrieve any message that is encoded into an image, by using the decoding algorithm.

5.2.3) Experimental Results:

On the left-hand side is the input image, which was encrypted with the message ‘Hello World !!’
On the right-hand side, we can observe the output steganographic image, with the message hidden in it. If observed very closely, there are some minute differences between the input and the output image, due to the change in the least significant bits. If this was programmed to change the most significant bits, the number of changes in the output image would be considerably higher, making the image unusable hence failing the process of steganography. When the output image is passed through the decoder function, we get the message ‘Hello World !!’ retrieved successfully.



6. Conclusion

In the recent digital world of today, namely, 1992 to present, Steganography is being used all over the world on computer systems. Many tools and technologies have been created that take advantage of old steganographic techniques such as null ciphers, coding in images, audio, video, and microdot.

Steganographic technologies are a very important part of the future of Internet security and privacy on open systems such as the Internet. Steganographic research is primarily driven by the lack of strength in the cryptographic systems on their own and the desire to have complete secrecy in an open-systems environment.

The project serves to understand how steganography can be implemented to secretly transfer data through images. The importance of digital image processing and its wide range of applications is Knowledge of steganography is of increasing importance to individuals in the law enforcement, intelligence, and military communities. The LSB method of image steganography has been noted to be very advantageous, considering its ease of implementation and inconspicuousness. With the research, this topic is now getting we will see a lot of great applications for Steganography in the near future.

7. Bibliography

[1] Trivedi M C Sharma S and Yadav V K 2016 Analysis of several image steganography techniques in spatial domain: a survey. In; Second International Conference on Information and Communication Technology for Competitive Strategies (ICTCS '16). ACM. Article 84

[2] Petitcolas F A P Anderson R J and Kuhn M G 1999 Information hiding-a survey. Proc. IEEE. 87(7): 1062–1078

[3] Mazurczyk W and Caviglione L 2015 Steganography in modern smartphones and mitigation techniques. IEEE Commun. Surv. Tutor. 17(1): 334–357

[4] Johnson, Neil F. "Steganography." Http://www.jjtc.com/pub/tr_95_11_nfj/sec101.html. N.p., Nov. 1995. Web.

[5] Shikha, and Vidhu Kiran Dutt. "International Journal of Advanced Research in Computer Science and Software Engineering." Http://www.ijarcsse.com/. N.p., Sept. 2014. Web.

[6] Niels Provos, and Peter Honeyman. "Hide and Seek: An Introduction to Steganography." IEEE Security & Privacy Magazine, May-June 2013. Web.

[7] Bharat Sinha, "Comparison of PNG & JPEG Format for LSB Steganography", International Journal of Science and Research (IJSR), https://www.ijsr.net/search_index_results_paperid.php?id=29031501, Volume 4 Issue 4, April 2015, 198 - 201