Security of Digital Images using Steganography Techniques based on LSB, DCT and Huffman Encoding

Nitin Kanzariya 1, Ashish Nimavat 2 and Hardik Patel 3
1 P.G. Student of Computer Engineering, CCET, Wadhwan city, India
2 Department of Computer Engineering, CCET, Wadhwan city, India
3 Department of Information Technology, SBPT, Bhavnagar, India
{ 1 kanzariyanitin141088@gmail.com, 2 ashishnimavat07@gmail.com, 3 hardikpatel.growmore@gmail.com}

Abstract. In insecure communication, data hiding techniques have an important role to protect private/important information from unauthorized access. Steganography is a hiding technique that hides the secret information inside the digital medium in undetectable manner. We believe that a good steganography technique requires high embedding capacity without compromising security as well. In this paper, an image steganography method based on combination of LSB, DCT and Huffman Encoding is proposed. Two 8 bit gray level images are used, first one as a cover image and second as a secret image. Before embedding the secret image inside cover image, Huffman Encoding is applied on the secret image to reduce the payload and hence embedding capacity is increased. Then Discrete Cosine Transform (DCT) is applied to obtain DCT coefficients of cover image. DCT coefficients are used to select random pixel locations for hiding bits of Huffman codes. The stego image is a result of hiding each bit from Huffman code of secret image in Least Significant Bit of the cover image at random locations based on threshold. The experimental result shows that the proposed algorithm has high embedding capacity and good invisibility of secret data. PSNR value is used to compare the result with other existing Steganography approaches. In addition, high security is maintained since the secret image cannot be extracted without the knowledge of decoding rules, Huffman table and the Key Matrix.

Keywords: Steganography, Huffman Encoding, LSB, DCT, PSNR.

I. INTRODUCTION

Steganography refers to the art and science of “invisible” communication. The word Steganography is a Greek word, Steganos, which means covered or hidden and graphy means writing or drawing. Therefore, meaning of Steganography is hidden writing. Steganography is hiding technique in which secret information is hidden inside digital media. It based on assumption that if the feature is visible, the point of attack is evident [1]. Main goal of Steganography is to provide invisible (hidden) communication in such a way that secret information is not visible to the observer [2]. The digital media without secret message is called cover media and digital media with secret message is called stego media. That is unauthorized parties could not be
able to differentiate between cover-object, which does not contain any secret message and stego-object, which contains secret message in hidden form. Steganography can use as both legal and illegal interests, e.g., military may use it for protecting secret information while terrorists may use it for spreading terroristic information [3]. This information hiding technique has recently become very important in number of application areas such as, Digital audio, video, and pictures.

Steganography is similar to cryptography as both are used to protect secret information from unauthorized access. In cryptography, the secret message is converted into unreadable (meaningless) form, whereas Steganography hides the secret message within other normal looking media. Cryptography provides visible communication, so there are more chances of attack on the system, and then the attacker can read the secret message. While Steganography provides invisible communication, so breaking a steganographic system require the attacker to first detect that Steganography has been used. It is possible to combine Steganography with cryptography, means first encrypting message using cryptography and then hiding the encrypted message using Steganography [9]. Even if an attacker detects the message from the stego-object, they would still need the decoding key to decipher the encrypted message.

Steganography types can be classified based on the digital cover medium [4], [14]. Steganography can be classified in three major types as (i) Text Steganography, (ii) Image Steganography and (iii) Audio/video Steganography. In Steganography, images are the most popular cover objects. Digital Image is most widely used on internet and provides high redundant data [5]. Different file formats such as jpg, png, bmp, gif are available for representation of digital image [6]. There are number of applications for digital image Steganography, such as copyright protection, secret communication, medical imaging and for data hiding in countries where cryptography is banned [5].

Organization: This paper is organized into following sections. Overview of related work is discussed in section II. Definition of performance analysis, proposed embedding procedure and extraction procedure are discussed in section III. The Embedding algorithm and recovery algorithm are discussed in section IV. Performance analysis and comparison with existing techniques is discussed in section V.

II. RELATED WORK

In Spatial domain Steganography, secret information is directly embedded into cover image. Least significant bits (LSB) replacement is a spatial domain technique [2], [13]. In LSB replacement, MSBs bits of the secret information are replaced with LSBs of pixels of the cover-image [11]. LSB Replacement provides high embedding capacity. Modifying the LSBs of cover image does not result in human-detectable difference because the amplitude of change is small.

Transform domain is also known as frequency domain, here image is first transformed using transformation such as the Discrete Cosine Transform (DCT) [5], [8],[10],[11] Discrete Fourier Transform (DFT) [5],[11] or Discrete Wavelet Transform (DWT) [5],[7],[8],[11] and then the message is embedded in transformed image. These techniques hide secret message in more significant regions of the cover image, making it more robust but provides less embedding capacity.

Amitava Nag [2] proposed an Image steganography based on LSB using X-box mapping in which authors used several Xboxes having unique data. The embedding was done by using four unique X-boxes with sixteen different values (represented by 4-bits). Each value was mapped to the four LSBs of the cover image. This mapping provides sufficient security to the secret image because without knowing the mapping rules no one can extract the secret image.

III. PROPOSED MODEL

This section includes definitions related to performance analysis followed by discussion of proposed model.

A. Definitions related to Performance Analysis

Capacity. The capacity is represented as size of secret information proportional to the size of stego image. It is defined as bits per pixel (bpp) and in terms of percentage using Equation 1.

\[
\text{Capacity} = \frac{\text{No of bits per pixel in Secret image}}{\text{No of bits per pixel in Cover image}}
\]  

(1)

Mean Square Error (MSE). It is defined as the square of error between cover image and the stego image. It is used to measure the distortion in the image using equation 2.
\[
MSE = \frac{\sum_{i=1}^{all \ pixels} \sum_{j=1}^{all \ pixels} (CI(i,j) - SI(i,j))^2}{N \times N}
\]

Where \(CI(i,j)\) represent the pixel of the cover image.
\(SI(i,j)\) represent pixel of the stego image.
\(N \times N\) represent the cover image size.

**Peak Signal to Noise Ratio (PSNR).** PSNR is the measurement of quality between the cover image and stego-image. It is measured in db using equation 3.

\[
PSNR = 10 \times \log(255^2 / MSE)
\]

**Correlation Coefficient (CC).** CC is used to measure the similarity (identical) between cover image and stego image using equation 4.

\[
CC = \frac{\sum (Xi - Xm)(Yi - Ym)}{\sqrt{\sum (Xi - Xm)^2 \sqrt{\sum (Yi - Ym)^2}}}
\]

Where, \(Xi\) is the intensity of the \(i\)th pixel of cover image.
\(Xm\) is the mean intensity of cover image.
\(Yi\) is the intensity of the \(i\)th pixel of Stego image.
\(Ym\) is the mean intensity of stego image.

**B. Proposed Model**

The secret image is embedded into cover image using LSB, DCT and Huffman Encoding to generate stego image providing more security, large embedding capacity and high PSNR.

**Least significant bit replacement.** Least significant bit (LSB) replacement is a common and simple approach to embed information in an image [13]. In this method the LSB(s) of a pixel is replaced with bit(s) from secret message or an image [11].

For example a bit pattern for 9 pixels of an 8-bit gray image can be as follows:

\[
(00101101 \ 00011100 \ 11011100) \\
(10100010 \ 11000100 \ 00001100) \\
(11010010 \ 10101101 \ 01100011)
\]

When the decimal number 201, for which binary representation is 11001001, was embedded into the LSBs of this part of an image, the resulting bit pattern should be as follows:

\[
(00101101 \ 00011100 \ 11011100) \\
(00101101 \ 00011101 \ 11011100) \\
(10100010 \ 11000101 \ 00001100) \\
(11010010 \ 10101101 \ 01100011)
\]

Although the 8 bits were inserted in first 8 pixels from trace of an image, only the 2 highlighted (underlined) bits need to be modified. On an average, only half of the pixel values in an image need to be modified while embedding the secret message. We have 256 possible intensities of gray scale image, changing the LSB of a pixel results in small changes in this intensity value [17]. Such modifications cannot be identified by the human eye, thus the message is hidden into the carrier successfully.

**Discrete Cosine Transform.** It transforms the image from spatial domain to frequency domain and separates the image into spectral sub-bands according to its visual quality, i.e. high, middle and low frequency components [8], [10]. Here DCT is applied on Cover Image to identify the pixel locations for hiding secret bits of Huffman codes. This provides randomness in insertion of secret data into the cover image.

**Huffman Encoding.** Huffman encoding is a variable length lossless compression technique and applied to any entity which represented in digital form. First secret image is encoded using Huffman coding and then resulting Huffman codes are embedded into cover image [15], [16]. Huffman codes are optimal codes that map one symbol of cover image to one code word [1]. Fig 1 show the block diagram of Huffman encoding in which secret image is converted to a 1-D bits stream. Huffmanantable (HT) represents binary codes of each symbol of cover image [16]. The Huffman table used at encoder and decoder side must be same. Thus the Huffman table is required for decoding process along with stego image [15]. Huffman encoding is mainly used for the following three characteristics:

- **Lossless Compression:** It ensures the preservation of actual data while compressing it.
- **Increase the Security:** Huffman encoded bit stream does not discloses anything because to extract the exact meaning, the Huffman table is required.
Authentication: It provides authentication, as if any single bit changes in the Huffman coded bit stream, Huffman table will not be able to decode the data.

**Embedding Procedure.** The block diagram of embedding procedure is shown in Figure 2. Before embedding, secret image was encoded using Huffman coding. The pixel intensities of an image are treated as different symbols and will vary from 0 - 255. 0 denotes minimum and 255 denotes maximum intensity. First step is to find the unique symbols and their probability from secret image. Then Huffman table is obtained from secret image using unique symbols and their respective probabilities. Huffman Table represents binary codes of each symbol and corresponding Huffman Codes. Huffman code is a prefix and minimum-length code in the sense that no other encoding has a shorter average length. Huffman encoding reduces the number of bits per pixel. Table 1 show that if secret image of size 20 x 20 is embedded inside cover image without compression, 3200 bits have to be embedded, while using Huffman encoding only 1253 bits have to be embedded. Thus Huffman encoding increases the embedding capacity. Then 2D-DCT is applied on the cover image to obtain the DCT coefficients. DCT coefficient value is used to identify potential pixels in cover image. The DCT coefficient values are generated in the form of matrix that is equal to the size of the cover image. For example original cover image show in figure 4. The trace of DCT coefficient matrix generated for the cover image is shown in the figure 5. Now pixels having DCT coefficients value lower than the threshold value are used for embedding 1-D bit stream of Huffman codes. The threshold value is empirically determined from the set of cover images. Here threshold value is taken as zero and hence the pixels with DCT coefficient value below zero are used for embedding the 1-D bit stream of Huffman codes that are highlighted in fig 5. The pixels in the cover image satisfying the threshold condition are not in sequence but distributed randomly in the cover image that is shown in fig 7. Generate key matrix based on DCT coefficient value show in fig 6 that is used to represents locations of potential pixels.

At the receiver side, key matrix is used to detect the locations of the potential pixels. These random pixels are called potential pixels. The randomization increases the security of the system and makes guessing difficult for unauthorized access. Once the potential pixels are found, 1-D bit stream can be embedded using LSB method. In embedding process, 1 MSB of 1-D bits stream is embedded in the 1 LSB of the potential pixel of cover image using LSB Replacement method.

---

**Fig 1: The block diagram of Huffman Encoding**

**Fig 2: The block diagram of embedding Process**

**Fig 3: The block diagram of extracting Process**
### TABLE I. NUMBER OF BITS WITH AND WITHOUT HUFFMAN ENCODING

<table>
<thead>
<tr>
<th>Secret Image size</th>
<th>Without Huffman Encoding</th>
<th>With Huffman Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No. of bits</td>
<td>Bits / pixels</td>
</tr>
<tr>
<td>20 X 20</td>
<td>3200</td>
<td>8</td>
</tr>
<tr>
<td>60 X 60</td>
<td>28800</td>
<td>8</td>
</tr>
<tr>
<td>100 X 100</td>
<td>80000</td>
<td>8</td>
</tr>
<tr>
<td>140 X 140</td>
<td>156800</td>
<td>8</td>
</tr>
</tbody>
</table>

**Recovery Procedure:** The block diagram of recovery procedure is shown in Figure 3. Recovery procedure requires stego image, Huffman Table and the key matrix which have to be shared between sender and the receiver. At receiver side, the key matrix is used to detect the locations of the potential pixels. Then 1 LSB is extracted from each potential pixel, which gives 1-D bit stream of Huffman codes. The 1-D bit stream is decoded using Huffman table to get original secret bits. Once the Decoded bits are obtained, they need to be arranged in a sequence so that the secret image can be generated. The Decoded bits are arranged based on the size of the secret image. Perfect arrangement of extracted bits generates an estimate of the secret image.

![Fig. 4: Original Cover Image.](image)

![Fig. 5: DCT coefficient values.](image)

![Fig. 6: Key Matrix with Threshold 0.](image)

![Fig. 7: Cover Image with Potential pixels.](image)

### IV. PROPOSED ALGORITHM

**A. Problem definition**

To embed secret image into cover image using LSB, DCT and Huffman Encoding to achieve secure communication.

**B. Assumptions**

I. An ideal channel is used to transfer the stego image.

II. Huffman Table and Key matrix are also transmitted over an ideal channel.

**C. Embedding Algorithm**

**Step 1:** Read both the Cover Image and the Secret Image.

**Step 2:** Find Huffman table for secret image.
Step 3: Apply Huffman encoding technique using Huffman table found in Step-2 on Secret Image.
Step 4: Apply DCT on the cover image using equation Eq (5).

\[
DCT_{ij} = a_{ij} \sum_{m=0}^{m-1} \sum_{n=0}^{n-1} C_{mn} \cos \frac{m(2m+1)i}{2m} \cos \frac{n(2n+1)j}{2n}
\] (5)

Where, \(i = 0, 1, 2 \ldots \ldots m-1\) and \(j = 0, 1, 2 \ldots \ldots n-1\)

\[
a_i = \begin{cases} 
\frac{1}{\sqrt{m}}, & i = 0 \\
\frac{1}{\sqrt{2/m}}, & 1 \leq i \leq m-1 
\end{cases}
\]

\[
a_j = \begin{cases} 
\frac{1}{\sqrt{n}}, & j = 0 \\
\frac{1}{\sqrt{2/n}}, & 1 \leq j \leq n-1
\end{cases}
\]

Step 5: Set threshold value.
Step 6: Track each pixel in Cover Image till end of 1-D bits stream of Huffman codes.

**Step 6.1:** If DCT coefficient value of pixel in cover image is less than threshold value, then replace LSB of pixels in cover image with next bit from 1-D bit stream of Huffman codes.

**Step 6.2:** Set value to 1 for that location in the key matrix.
Step 7: Stego Image with visible features similar to Cover Image is generated.
Step 8: End.

**D. Recovery Algorithm**

Step 1: Extract the least significant bits of each potential pixel of the stego image to get the bit stream of Huffman codes using key matrix.
Step 2: Decode the bit stream of Huffman codes that are extracted in Step-1 using the Huffman table.
Step 3: Construct the Secret Image by arranging bits according to its size.
Step 4: End.

**V. PERFORMANCE ANALYSIS**

Several images such as Barbara, peppers, jet, Lena and boats are selected as a cover images and secret images as shown in Figure 8. Different size and formats have been used for Secret Image for the purpose of performance analysis of proposed method. The secret cameraman image is embedded in the cover image Lena using proposed algorithm to generate stego image as shown in Figure 9. The quality of stego image is same as cover image by appearance and also by statistical characteristics. Table 2 shows various performance metrics. Here secret image (cameraman.gif) is embedded into cover image (Lena.jpg) (512x512). The value of PSNR decreases as capacity increase, while PSNR between original secret image and extracted secret images is infinite. It means both images are 100% Identical.
Table 3 shows PSNR Variations for different Secret image formats with cover image Lena.jpg (512 x 512). It is observed that the secret image formats does not affect the quality of stego image.

Table 4 shows PSNR and Correlation coefficient for different cover images having same secret image size to be embedded. It is observed that different cover image has minor change on the quality of stego image as PSNR values are nearly same.

Table 5 shows the comparison of PSNR value between existing methods and our proposed algorithm. It is observed that PSNR is high for our proposed algorithm as compared to existing algorithms.
VI. CONCLUSION

The Steganography is hidden communication to protect confidential information. In this paper, a novel image steganography approach based on LSB, DCT and Huffman Encoding is proposed. DCT provides randomization and Huffman encoding provides high embedded capacity. The algorithm improves the security and the quality of the stego image. Result shows that the proposed method is better in compare to other existing methods. According to the results, the stego images are almost identical to the cover images and it seems very difficult to differentiate them. The secret image recovered from stego image was also found identical to the original secret image. PSNR depends on size of secret image. We have used different size for secret image and have obtained high PSNR compared to some of the existing methods.

REFERENCES