Robust Lossless Semi Fragile Information Protection in Images

Pushkar Dixit[1], Nishant Singh[2], Jay Prakash Gupta[3]

[1]Faculty of Engineering and Technology Agra College, Agra, India
[2]Poornima Institute of Engineering and Technology, Jaipur, India
[3]Infosys Limited, Pune, India

Abstract **– Internet security finds it difficult to keep the information secure and to maintain the integrity of the data. Sending messages over the internet secretly is one of the major tasks as it is widely used for passing the message. In order to achieve security there must be some mechanism to protect the data against unauthorized access. A lossless data hiding scheme is proposed in this paper which has a higher embedding capacity than other schemes. Unlike other schemes that are used for embedding fixed amount of data, the proposed data hiding method is block based approach and it uses a variable data embedding in different blocks which reduces the chances of distortion and increases the hiding capacity of the image. When the data is recovered the original image can be restored without any distortion. Our experimental results indicate that the proposed solution can significantly support the data hiding problem. We achieved good Peak signal-to-noise ratio (PSNR) while hiding large amount of data into smoother regions.**

Keywords **– Data Security, Lossless Data Hiding, Semi Fragile, Encryption, Decryption, Internet Security, Steganography, Information Protection**

I. Introduction

ONE of the most important issues arising out of the wide spread use of internet is the security of information. Cryptography has long been used to maintain the secrecy of the data. There are a number of algorithms to encrypt and decrypt a message. However, sometimes it is not enough to keep a secret message. It is essential that except for the intended receiver of the message, no one should even come to know that any communication is taking place.

Nowadays there has been a rapid development in the internet and its technology, the individual prefers internet as the primary medium for communication between one parts of the world to another. There are many possible ways to transmit the data across the internet: via e-mails, chats, video streaming, video calling, etc. Internet has made the transmission of the data very simple, fast and accurate. Internet has both its merits and demerits as the intended users can view the information and one who is not intended can also do. Thus in order to make it visible only to the intended users, we must have some method such that people who are not intended may be prevented from viewing information directly. Thus we can say that one of the main problems with sending the data over the internet is 'security threat' posed, in other words personal or confidential data can be stolen or hacked in many ways. Therefore it becomes very important to take data security into consideration. It is one of the most essential factors that need attention during the process of sending or receiving of data.

Before proceeding further it is necessary to understand the conceptual difference between cryptography and steganography. Cryptography conceals information by encrypting it into cipher text and transmitting it to the user using an unknown key, whereas steganography hide the cipher text into a seemingly invisible image or other formats. The word steganography is of Greek origin which means "covered or hidden writing" [1]. Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. It is therefore a book on magic. It is emerging in its peak because it does not attract anyone by itself.

Encrypting data into some form has been the most popular approach for protecting information, but this protection can be breached with enough computational power. An alternate approach to encrypting data is hiding it by making this information appear to be something else. This way only intended user can receive a true content. In particular, if the data is hidden inside an image then everyone but our intended users or the person it is meant for can view it as a picture which is transmitted. At the same time he/she could still retrieve the true information while the unintended people would view it only as an image.

Data hiding has its application in various areas due to the image being the most common digital media transmitted over the internet. Thus practically, it is very difficult for an unwanted user to masquerade the information that is transmitted over the channel by checking each and every image as it is a very time consuming job. Thus it is quite a nice option to choose images to hide the data and send them over internet and the receiver can easily extract the information from image.

DOI: 10.9781/ijimai.2014.269

Two important properties of steganographic technique are perception and data hiding capacity. Steganography generally exploits human perception because human senses are not trained to look for file that has hidden information inside it. Therefore steganography disguises information from people who try to hack it. Data hiding capacity is the amount of information that can be hidden in the cover object. The cover object means the image that we use for embedding the data and the stego object means the image obtained after embedding the data into cover object.

The different types of steganography techniques are substitution, transform domain, spread spectrum, statistical and distortion techniques and cover generation techniques. Substitution techniques replace the least significant bit of each pixel in the cover file with bits from the secret document. The transform domain techniques hide secret information in the transform space (like frequency domain) by modifying the least significant coefficient of the cover file. Most of the research work done in the area of transform domain embedding is focused on taking the advantage of redundancies in Discrete Cosine Transformation (DCT). Spread spectrum techniques spread hidden information over different bandwidths. Even if the parts of the message are removed from several bands, there would still be enough information present in other bands to make out the message. Statistical techniques change several statistics of the cover file and then split it into blocks where each block is used to hide one message bit. The most obvious limitation to these techniques is that the cover image must be very largely compared to the secret information it is going to carry fixed payload over the image. We can hide large amount of information in multiple files but it could lead to suspicion. Therefore, it is very important to use only one image file to hide the entire secret information.

Each application using information hiding technique has different requirements depending on the purpose of the application. Generally, there are four issues that we encounter while designing the algorithm: perceptibility, payload, security, robustness, and they are common to most applications. Because there are tradeoffs existing between those requirements, it is very challenging to design an algorithm that satisfies all the four requirements.

Hiding a fixed amount of data in the image may give a uniform distribution of data but it makes the image more suspicious for the attackers as the changes in the image are visible. We need to check the image by the means of threshold and make sure it is able to adapt to the amount of data that we want to embed into it. If we do not perform this step then it may result in loss of information or poor embedding of data. Also we have to select such an image which does not have large sharp details.

We present a simplified embedding algorithm based on difference expansion, which is capable of minimizing the distortion of the stego-image presents in the traditional difference expansion algorithms. The main principle underlying the proposed framework is blocks and centralized difference expansion. In the framework, the original cover image is partitioned into continuous non-overlapping blocks. The bits embedded in each block depend on its block size and the image complexity. A new approach is employed to find the image complexity of each image block, and all the blocks are categorized into three levels according to their block intensity values. Finally varying amounts of data are assigned to image blocks at different intensity levels. Although there are three types of blocks in the embedding procedure, only 1 bit is required to record these three blocks. This way, the proposed method can reach a higher hiding capacity while maintaining good visual quality of the stego-image. Our major concern is that the image should not show any visual effects and it carries as much data as possible. Some other approaches can also be utilized with to enhance the algorithm [2-9].

This paper is outlined as follows: section 2 provides some of the core concepts used in image steganography and presents a survey of efforts done by researchers in the past to address this problem; section 3 describes the framework for the scanning the image as well as embedding and extracting the data; section 4 discusses the result and compare it with classical approach; finally section 5 summarizes the novelties, achievements and limitations of the proposed solutions and indicate some future directions.

II. LITERATURE REVIEW

In this section, we focus on the previous work done by several researchers in the area of data hiding, steganography and steganalysis. Data hiding and steganography can be seen as instances of the image security. People have been resorting steganography or information hiding since Greek times. However, digital steganography is a relatively new research field. Since being undetectable is one of the essential requirements for steganographic applications, steganography and steganalysis techniques are evolving in competition with each other.

The aims of improving the original DE (data embedding) proposed by researchers are twofold: first is to make the embedding capacity as high as possible, second is to make the visible distortion as low as possible. To achieve high embedding capacity, the reviewed schemes adopted three different approaches: (i) simplifying the location map in order to increase its hiding capacity, (ii) embedding payload without location map, and (iii) expanding differences more than once which allows more data to be embedded. Meanwhile, the visual quality may be enhanced by: (i) using a predefined threshold T, (ii) selecting smooth areas to embed data, and (iii) using sophisticated classification functions. However, there is a tradeoff between distortions and embedding capacity. If distortion is minimized, lesser data can be embedded. On the other hand, if the embedding capacity is increased, it results in low visible quality.

Most of the researchers in the field of data hiding or image steganography have considered capacity and robustness as a key for their approach. Some of them have considered both and some of them have considered them individually. In most of the techniques, fragile images are used which is of no use after the extraction of data and it can't be restored to the original state. The major part of the research attention has been paid to the perception part of the topic rather than that of capacity.

Recently, Li et al. [10] proposed a reversible data hiding (RDH) scheme based on two-dimensional difference-histogram modification by using difference-pair-mapping (DPM). First, by considering each pixel-pair and its context, a sequence consisting of pairs of difference values is computed. Then, a two-dimensional difference-histogram is generated by counting the frequency of the resulting difference-pairs. Finally, reversible data embedding is implemented according to a specifically designed DPM. Where, the DPM is an injective mapping defined on difference-pairs. It is a natural extension of expansion embedding and shifting techniques used in current histogram-based RDH methods.

Faragallah [11] proposed quadruple difference expansion-based reversible data hiding method for digital images which is characterized by two aspects. First, reversible data hiding scheme is enhanced to exhibit data hiding in color palette images. Second, the embedding level is improved by using quadruple difference expansion to guarantee the embedding of 2-bit data into color images. But they have not considered the level of the details present in the image and hide 2-bit at each place (i.e. smoother and non-smoother ares) which is a drawback of this approach.

In the this section, we discuss several approaches used by researchers [12-22] with the aim of being aware to the latest research carried out our focus is on those related to the formulated problems in this paper.

A. Literature Survey

The word steganography is originally derived from Greek words which mean ''Covered Writing''. It has been used in various forms for thousands of years. In the 5th century BC, Histaiacus shaved a slave's head, tattooed a message on his skull and the slave was dispatched with the message after his hair grew back [23-25]. In Saudi Arabia at the King Abdulaziz City of science and technology, a project was initiated to translate some ancient Arabic manuscripts into English on secret writing which are believed to have been written 1200 years ago. Some of these manuscripts were found in Turkey and Germany [26].

Color palette based steganography exploits the smooth ramp transition in colors as indicated in the color palette. The LSBs here are modified based on their positions in the palette index. Johnson and Jajodia [23] were in favour of using BMP (24 bit) instead of JPEG images. Their next-best choice was GIF files (256-color). BMP as well as GIF based steganography apply LSB techniques, while their resistance to statistical counter-attacks and compression are reported to be weak. BMP files are bigger as compared to other formats which render them improper for network transmissions. However JPEG images were avoided at the beginning because of their compression algorithm which does not support a direct LSB embedding into the spatial domain.

One of the earliest methods to discuss digital steganography is credited to Kurak and McHugh [27]. They proposed a method which resembles embedding into the 4 LSBs (least significant bits). They also examined image downgrading and contamination which is now known as image based steganography. Provos and Honeyman [24], at the University of Michigan, scrutinized three million images from popular websites looking for any trace of steganography. They have not found a single hidden message. Embedding hidden messages in video and audio files is also possible. Examples exist in [28] for hiding data in music files, and even in a simpler form such as in Hyper Text Markup Language (HTML), executable files (.EXE) and Extensible Markup Language (XML) [29].

Vleeschouwer et al. [30] solved the problem of salt-and-pepper noise artifact by using a circular interpretation of bijective transformation. The proposed algorithm guarantees the coherence of the transformation interpretation and, consequently, ensures total reversibility. To improve the performance of Fridrich et al.'s scheme in terms of message bits, Celik et al. [31] presented a high capacity, low distortion reversible data embedding algorithm by compressing quantization residues. Images can be obtained after a quantization process and then the CALIC lossless compression algorithm is used to get the compressed residues. The remainder of the compression space is used to hide the secret message. In addition, Ni et al. [32] utilizes zero or minimum point of histogram. If the peak is lower than the zero or minimum point in the histogram, it increases pixel values by one higher than the peak values to lower than the zero or minimum point in the histogram. While embedding, the whole image is searched thoroughly. Once a peak-pixel value is encountered, if the bit to be embedded is '1' the pixel is added by 1, else it is kept intact. The algorithm essentially does not follow the general principle of lossless watermarking. The advantages of this algorithm are (i) it is simple, (ii) it always offers a constant PSNR 48.0dB, (iii) distortions are quite invisible, and (iv) its capacity is high. The disadvantage is that the algorithm is time consuming because it searches the image several times.

Tian suggested multiple-layer embedding in order to achieve larger embedding capacity [33]. For example, the second layer embedding would take place in the orthogonal direction, where the difference image is obtained by performing integer Haar wavelet transform on the embedded image in column direction. If the capacity of the two-layer embedding is still insufficient for the payload, a third layer embedding is needed. One performs integer Haar wavelet transform in row direction again and repeats the embedding operation. Such a process continues until the total embedding capacity is large enough for the payload. However, multiple-layer embedding results in some unexpected problems. First, image quality (in terms of peak signal-to-noise ratio (PSNR)) drops greatly after the first layer embedding due to the use of large differences. Second, the new difference image has smaller embedding capacity than its predecessor. Each layer-embedding progressively decreases the correlation not only in the embedding directions but also of the neighborhood.

In [34], a lot of secret data bits are hidden in a vector. After the difference of that vector is expanded the difference expansion by generalized integer transform to make it work for more than two pixels per vector with k-1 bits of secret data hidden into k pixels. However when it is pixel pair difference expansion or difference expansion for more than two pixels, there is additional information to save this keeps track of the characteristics of a vector. In this case, a location map is needed because it records the characteristics of a vector.

Maniccam and Bourbakis [35] presented a lossless image compression and information hiding scheme. In their methodology, they have performed both lossless compression and encryption schemes which are based on known SCAN patterns generated by the SCAN methodology. This SCAN is a formal language based two-dimensional spatial accessing methodology which can efficiently specify and generate a wide range of scanning paths or space filling curves. This algorithm has lossless image compression and encryption abilities. The only advantage of simultaneous lossless compression and strong encryption makes the methodology very useful but the drawback of the methodology is that compression-encryption takes longer time.

Paulson [36] reported that a group of scientists at Iowa State University were focusing on the development of an innovative application which they call ‘‘Artificial Neural Network Technology for steganography (ANNTS)’’ aimed at detecting all present steganography techniques including DCT, DWT and DFT. The inverse discrete Fourier transform (iDFT) encompasses round-off error which renders DFT improper for steganography applications.

Abdelwahab and Hassaan [37] proposed a data hiding technique in the DWT domain. Both secret and cover images are decomposed using DWT (1st level). Each of which is divided into disjoint 4 X 4 blocks. Blocks of the secret image fit into the cover blocks to determine the best matches. Afterwards, error blocks are generated and embedded into coefficients of the best matched blocks in the HL of the cover image. But the extracted payload is not totally identical to the embedded version as the only embedded and extracted bits belong to the secret image approximation while setting all the data in other sub images to zeros during the reconstruction process.

In [38], authors used a spatial domain technique in producing a finger print secret sharing steganography for robustness against image cropping attacks. The logic behind their proposed work is to divide the cover image into sub-images and compress and encrypt the secret data. The resulting data is then sub-divided in turn and embedded into those image portions. To recover the data, a Lagrange Interpolating Polynomial is applied along with an encryption algorithm. The computational load was high, but their algorithm parameters, namely the number of sub-images (n) and the threshold value (k) were not set to optimal values leaving the reader to guess the values. Data redundancy that they intended to eliminate does occur in their stego-image.

Lin et al. [39] created a method to restore the marked image to its pristine state after extracting the embedded data. They achieved this by applying the pick point of a histogram in the difference image to generate an inverse transformation in the spatial domain. The example shown in their hiding phase section might not be sufficient to verify the accuracy of the algorithm. Some questions remain unanswered such as what happens when we have two peak points instead of one? On which criterion will we base our selection? It is very likely that after the subtraction process we will have some values that collude with the peak value which confuses the extraction of the embedded data.

Wu and Shih [40] presented a GA-based algorithm which generates a stego-image to break the detection of the spatial domain and the frequency-domain steganalysis systems by artificially counterfeiting statistical features. This is the first paper of utilizing the evolutionary algorithms in the field of steganographic systems. Time complexity, which is usually the drawback of genetic based algorithms, is not discussed in this paper. They have only mentioned that the process is repeated until a predefined condition is satisfied or a constant number of iterations are reached.

Raja et al. [41] used wavelet transforms that map integers to integers instead of using the conventional wavelet Transforms. This overcomes the difficulty of floating point conversion that occurs after embedding. Some other approaches also can be employed to improve the performance [42-48].

III. Proposed Methodology

A good image steganography approach aims at concealing the highest amount of data in an image while maintaining its imperceptibility so that its visual quality is not hampered or least affected. The least significant bit scheme is one of the simplest and easily applicable data hiding methods, where bits of secret data are directly embedded in the least significant bits of each image pixel. In traditional data embedding schemes, the exact original image cannot be recovered after data embedding. Compared with loss prone embedding methods, reversible data embedding methods embed a fixed payload into a digital content in a reversible fashion. After embedding, the image changes very little or looks no different.

Another obvious feature of reversible data embedding is the reversibility, that is, when the digital content has been used for the purposed it was embedded, one can extract the embedded data and restore the original content. There are a number of challenges that must be addressed to perform data hiding in images. The issues that we must keep in mind while designing the algorithm are perceptibility, payload, security and robustness. We must maintain a trade-off among all these and find a better solution to the problems encountered in data hiding.

Steganography techniques aim at secretly hiding data in a multimedia carrier such as text, audio, image or video, without raising any suspicion of alteration to its contents. The original carrier is referred to as the cover object. In this work, we mainly focused on image steganography. Therefore, the term cover object now becomes cover image. Figure 1 illustrates a basic information hiding system in which the embedding technique takes a cover image and a secret image as inputs and produces as output a stego image. Receiver side carry out the extraction process to retrieve the secret message from the stego image sent over the communication links to the receiver.

The proposed approach is comprised of three steps as shown in Figure 1. In the first step secret message is generated and an image is being selected. If the image is too small for the data then another image is selected to make sure that the data gets embedded into the image. In the second step data embedding process is carried out. In the last step data is being extracted by the extraction algorithm and the original image and protected information are recovered.

A. Message generation and image selection

The basic step in data hiding is that first of all we should have a data or secret message to be hidden in any form i.e. it may be in the form of text or any other form. After getting the data we change its form to some digital form (i.e. binary). The algorithm to convert the message into binary array form is as follows:

1. Read and store the characters of a message in an array A.
2. Do for 1 to length of A
   a. Convert each character into its decimal value.
   b. Convert those values from decimal to binary value.
   c. Store these values in an array A’.
3. Store the length of A’ in L.

After we successfully converted the secret message into binary form, we know the total number of bits of the message that we need to embed into the image. Next we select an image and scan it to calculate the amount of data it can carry. If the message data is larger than the hiding capacity of the image then select different image otherwise continue with the same. The image that we are using here is called the cover image and the image obtained after embedding the data is known as the stego-image. Now, we have the data and the image in which the data is to be embedded.

B. Data embedding algorithm based on Difference Expansion

There are lots of redundancies in a digital image. If we change some pixel values to some extent, the appearance of the picture is similar to the original one. So, data embedding can be obtained by changing some pixel values in an image. The DE technique [33] reversibly embeds one bit data into two integers, which is explained in Figure 2. The DE technique uses the difference between two pixel values to embed one bit. Assuming there are two grayscale values x = 206, y = 201, we reversibly embed one bit b = 1. First the integer average and difference of x and y are computed,

$$l = \lfloor (x+y)/2 \rfloor \tag{1}$$

$$h = x - y \tag{2}$$

where the symbol ⌊⌋ is the floor function meaning "the greatest integer less than or equal to". Next the difference value h is represented in its binary representation.

$$h = 5 = 101_2 \tag{3}$$

Then embedding bit b is appended into the binary representation of h after the least significant bit (LSB), and the new difference value h' is obtained.

$$h' = 101b_2 = 1011_2 = 11 \tag{4}$$

Mathematically, this is equivalent to

$$h' = 2 * h + b = 2 * 5 + 1 = 11 \tag{5}$$

Finally the new grayscale values are computed, based on the new difference value h' and the original integer average value l, and new two pixel values x = 209, y = 198 are obtained. After finishing this process, one bit is embedded into the two pixel values.

$$x' = l + \lfloor (h'+1)/2 \rfloor = 203 + \lfloor (11+1)/2 \rfloor = 209 \tag{6}$$

$$y' = l - \lfloor h'/2 \rfloor = 203 - \lfloor 11/2 \rfloor = 198 \tag{7}$$

This method is focused on the data rather than the image as integrity of data is more important. In this technique, we have a text message that is to be hidden in the image and an image that is available in which the data is to be embedded. We consider that the image is of size ‘M x N’. We use the image in our computations as grey scale image. Further we need to consider a block of particular size let it be ‘m x n’. We also

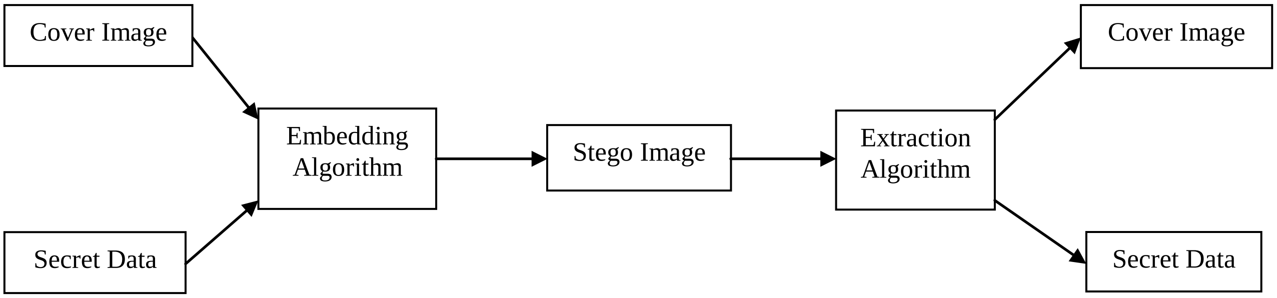


Fig. 1. Data Hiding and Extraction Process

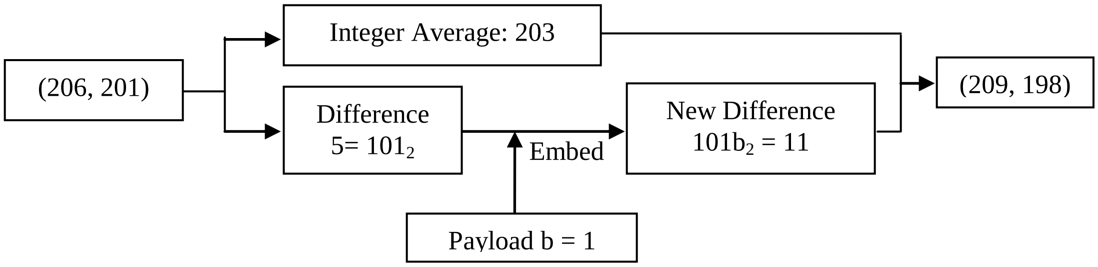


Fig. 2. An example of data embedding by Tian (2003) [33]

select a threshold value 'T' for keeping the image distortion free.

The image is divided into various non-overlapping blocks of size 'm x n'. For each block we have various components in the block. We arrange them in increasing order, select the mid value and subtract all the values from this mid value. Some values give us positive result some give negative. In the next step, we select the maximum difference of all the values and compare it with a threshold in order to decide the amount of data that we can embed into that block. The blocks are named as a, b and c. if a block belongs to type 'a', it means this block is located in a rather smooth area since the difference values are all very small. In this case, it is a very suitable block to hide more secret data bits here 3 bits of data may be embedded. Block 'b' belongs to the region of sharp detail or edges which can acquire only 1 bit of data and in the last block 'c' no secret data is embedded as to avoid distortion. We also use a record bit in order to identify the block after the embedding thus we set the record bit as 1 for all data embedded into block 'a' and record bit '0', for all data embedded into block 'b'. For block 'c', it is not needed. The input to this algorithm is an image and a secret message binary array threshold value. The output is the marked stego image. The process is carried out as shown in Figure 3.

1) Secret Data Embedding Algorithm

1. Segment the cover image into non overlapping blocks of size m x n
2. Label the components as $v_0,v_1,v_2,\dots\dots\dots v_k$ where k= mn-1
3. Find v_{mid} and hence compute differences as $d_0,d_1,d_2,\dots\dots\dots d_{m-1}, d_{m+1},\dots. d_{k-1}$
4. Define threshold 'T' and 'd_{max}'.
5. Find the type of block (i.e a, b and c) using T and divide the image into smooth regions and edge regions on the basis of threshold and difference in pixel values.
6. We embed large data in smooth regions and less data in non-smooth regions. The conditions are as follows:
   a. If $d_{max} < T/8$, then it belongs to block 'a' and three bits can be embedded into each block with record bit being 1.
   b. If $T/8 \le d_{max} < T$, then it belongs to block 'b' and one bit can be embedded to each block with record bit being 0
   c. If $d_{max} \ge T$, then it belongs to the block 'c' and it is a non-embeddable block.
7. Output: an image with data embedded (i.e. stego image).

Data embedding process is illustrated in Figure 3 and can be summarized as follows: first of all non-overlapping blocks of a particular size are extracted from the image; then for a particular block, all the pixel values are sorted and mid value are subtracted from each pixel value. Maximum difference value is selected to decide the type of the block because the magnitude of the local differences can adequately describe the edges of the local regions of the image. According to the details of the edges using maximum difference value and a threshold value we decide the type of the bloc and further the number of bit to be embedded in that block. We are not embedding any information in that block which have very high level of edge details to avoid any degradation in the image.

C. Secret Data and Cover Image Extraction

In this method, now, we have a stego image containing secret information. We consider that this image is also of size 'M x N'. In the next step, we need to consider a block of particular size let it be 'm x n' which is same as it was for the embedding part otherwise we would not be able to extract the data from it. We have prior knowledge of threshold value 'T' and we use its same value here also. The image is divided into various blocks of size 'm x n'. For each block we have various components in the block. We arrange them in increasing order and select the mid value then subtract all the values from the mid values. Some values give us positive result while some give negative then we select the maximum difference of all the values and compare it with the threshold in order to decide the amount of data we can extract from that block. If a block belongs to type 'a', it means that this block is located in a rather smooth area since the difference values are all very small. In this case, it must contain 3 bits of concealed data in it. We check the record bit, if it is '1' we need to extract the data by subtracting the last 3 bit from the values of the component. Then with the help of the d_{max} we restore the image value to what it was earlier. On the other hand if the block is of type 'b' it contains only 1 bit of data and we extract this data in the same way as we extracted for the block 'a'.

In this situation we only need to extract the LSB (Least Significant Bit) and check the record bit being '0'. For the blocks of type 'c', there is no secret data embedded. The input to this phase is a stego-image and a threshold value and the output is original restored cover image and the secret data. During the extraction process we also notice that we are able to recover the original cover image same as it was before. Thus, this suggests that there is no noise further added during the embedding other than the secret data embedded (see Figure 4).

1) Secret Data Extraction Algorithm

1. Segment the stego image into non overlapping blocks of size m x n
2. Label components as v_0,v_1,……….$v_{k,}$ where k= mn-1
3. Find v_{mid} and hence compute differences as d_0,d_1,d_2,……….d_{m-1}, d_{m+1},…. d_{k-1}
4. We already know threshold 'T' and 'd_{max}'.
5. Find the type of block (i.e a, b and c) using T and divide the image into smooth regions and edge regions on the basis of threshold and difference in pixel values.
6. We extract large data from smooth regions and less from non smooth regions. The conditions are as follows
   a. If $d_{max} < T/8$, then it is block of type 'a' therefore three bits are to be extracted from each block with record bit being 1.
   b. If $T/8 \leq d_{max} < T$, then it is 'b' type block and one bit can be extracted from each block with record bit 0.
   c. If $d_{max} \geq T$, then the block belongs to type 'c' and no data can be extracted from such block.
7. Restore cover image with the extracted secret data.

The hidden information extraction approch is described in the Figure 4 and can be summrized as follows: first of all again the image is divided into non-overlapping blocks and for each block the pixel intensities are sorted and subtracted with the mid value in order to find the type of the block using the maximum difference value (i.e. the number of bits to be extracted from a particular block), it should be noted that the maximum difference value will be same as it was previously because intensity differences are invarint to the monotonic intensity change caused by the embedding bits; three, one or none number of bits are extracted according to the type 'a', type 'b' and type 'c' of the block respectively; and finally the information and image both are recovered.

Our approach is able to embed more data in smoother images keeping the distortion not visible at all. The most important fact as well as advantage of using this approach is that we are able to restore the original image back to its initial state (i.e. the state in which it was earlier before data embedding) thus our approach makes sure that no additional noise is added into the image other than the. This is one of the features of the DE (Difference Expansion) technique that the image can be restored to its previous state.

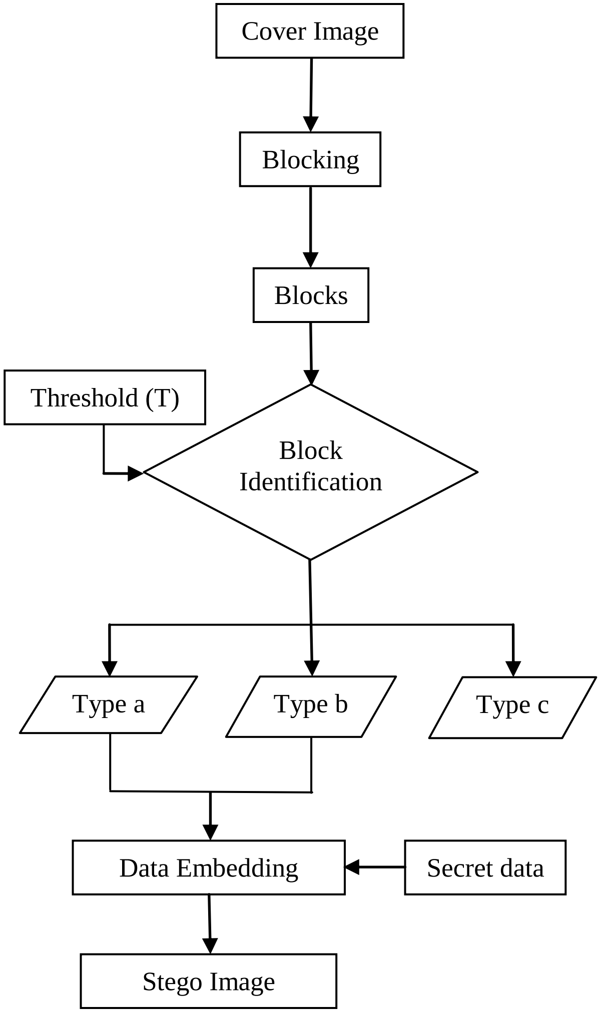


Fig. 3. Data Embedding Flowchart

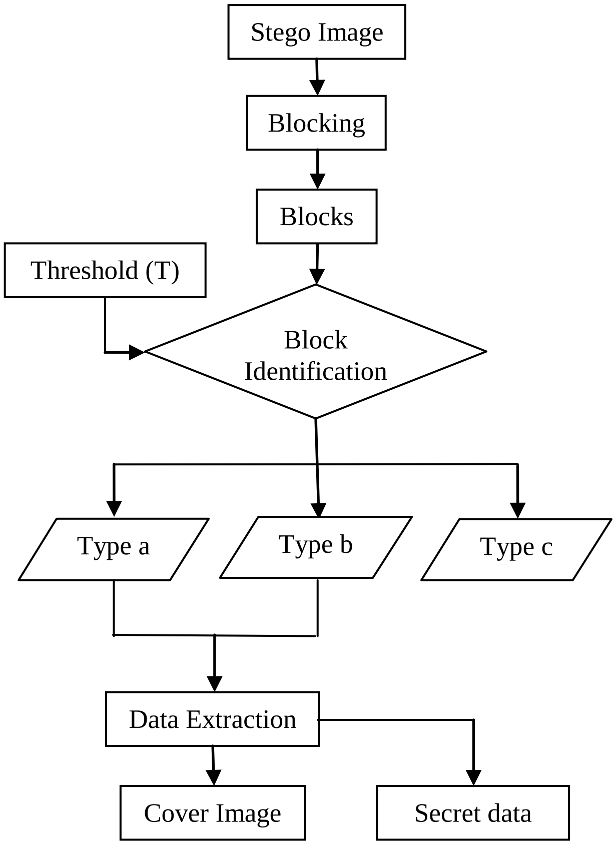


Fig. 4. Data Extraction Flowchart

IV. Results and Discussions

This section analyses the various aspects of the proposed method. Embedding is not prone to any visual attacks as the changes are hardly visible in the image. In addition to this we use an image that has smoother regions and hence it can acquire more data as compared to the one with less smooth region. Here we test our method for the correctness and performance. The PSNR (peak signal to noise ratio) is used to measure the quality of stego image when compared with the cover image. It is calculated using equation 8, where MSE is mean square error given by equation 9.

$$\text{PSNR} = 10 \log_{10} \frac{(2^n - 1)^2}{\text{MSE}} \qquad (8)$$

Where, MSE is given by

$$\text{MSE} = \frac{\sum (\hat{x} - x)^2}{A} \qquad (9)$$

The quality of the image is higher if the PSNR value of the image is high. Since PSNR is inversely proportional to MSE value of the image, the lower MSE value yields higher PSNR value. It means, the better the stego image quality the lower the MSE value will be. Using variable data to be hidden in the image we test the images for the PSNR and the Bit per Pixel (BPP). Where BPP is given by

$$\text{BPP} = \frac{n}{\text{MXN}} \qquad (10)$$

Where n is the number of bits to be embedded in the image while M and N are the dimensions of the image.

A. Data Set Used

In order to evaluate our proposed, we have used a data set of 20 images (i.e. 20 cover images). All the images used are gray scaled images of size 512 X 512. Figure 5 shows the 20 cover images used in this paper for the experiments. The simulation for the experiment was set up and carried out on a Windows XP Professional with 1.8 GHz dual core processor and 1 GB of RAM.

The proposed approach is implemented on the publicly available MATLAB R2009b. We have further referred the images of the data set as 01.jpg, 02.jpg, 03.jpg…..20.jpg in the Figure 5.

B. Experimental Results

We have obtained the stego image after data embedding (payload size of 61376 bit with BPP of 0.2341) in all cover images of the dataset and recovered all the images after extraction of data as it is shown in Figure 6. First column shows the original input cover images, second column depicts the stego image obtained after data hiding and third column shows the recovered images obtained after data extraction.

In Figure 6, we can see that there is nearly no visual difference among input, stego and recovered images it means our embedding algorithm is robust to noise and able to recover original image accurately. Figure 7 illustrates the quality of stego image with the size of payload data embedded. It shows the graph between PSNR and BPP for all images of the dataset used. From this figure it is clear that if size of the data increases (i.e. BPP increases) quality of the stego image degrades because the value of PSNR decreases.

We are able to achieve high embedding capacity with keeping PSNR in range and also the visual quality of the image is not getting too distorted to visualize as it is depicted by Table 1. This table listed the value of the PSNR obtained for each images after data embedding of varying sizes. The value of the PSNR is highest for 13.jpg in each case of payload data because 13.jpg has largest smoother area (we can see it in the 13th image of Figure 5) and the value of the PSNR is lowest for 14.jpg in each case of payload data because 14.jpg has lowest smoother area(we can see it in the 14th image of Figure 5).

(a) 01.jpg (b) 02.jpg (c) 03.jpg (d) 04.jpg (e) 05.jpg

(f) 06.jpg (g) 07.jpg (h) 08.jpg (i) 09.jpg (j) 10.jpg

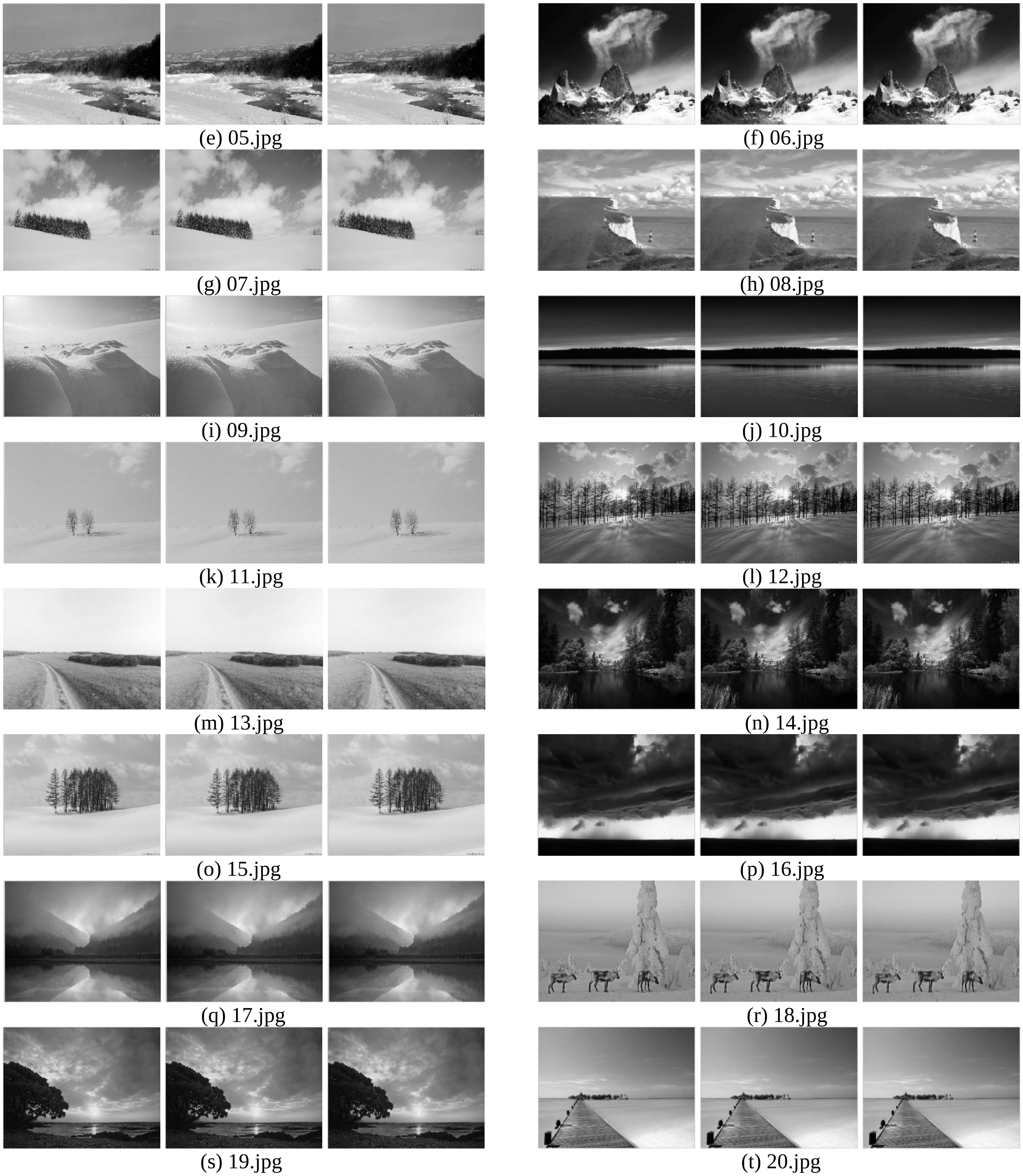
(k) 11.jpg (l) 12.jpg (m) 13.jpg (n) 14.jpg (o) 15.jpg

(p) 16.jpg (q) 17.jpg (r) 18.jpg (s) 19.jpg (t) 20.jpg

Fig. 5. 20 cover images of the data set used in this paper

(a) 01.jpg (b) 02.jpg

(c) 03.jpg (d) 04.jpg



(e) 05.jpg

(f) 06.jpg

(g) 07.jpg

(h) 08.jpg

(i) 09.jpg

(j) 10.jpg

(k) 11.jpg

(l) 12.jpg

(m) 13.jpg

(n) 14.jpg

(o) 15.jpg

(p) 16.jpg

(q) 17.jpg

(r) 18.jpg

(s) 19.jpg

(t) 20.jpg

Fig. 6. Cover image (1st column), stego image (2nd column) and restored image (3rd column) for input images.

TABLE I.
RESULTS AFTER TESTING ALL THE IMAGES OF THE DATA SET

Image	**Payload size in bits and their corresponding Bit Per Pixel for the images used for testing**					
	14320	**29896**	**45008**	**61376**	**75304**	**93568**
	.0546	**.1140**	**.1717**	**.2341**	**.2873**	**.3569**
01.jpg	43.76	40.24	38.36	36.92	35.70	34.38
02.jpg	43.26	39.76	37.36	36.35	35.59	33.53
03.jpg	43.56	40.37	39.36	37.17	34.63	33.55
04.jpg	43.91	40.57	38.55	36.78	35.53	34.15
05.jpg	42.54	39.42	37.59	36.24	35.37	34.31
06.jpg	43.08	38.66	36.69	35.21	34.26	33.42
07.jpg	42.71	39.38	37.72	36.49	35.56	34.40
08.jpg	45.87	42.37	38.99	37.04	35.88	34.57
09.jpg	44.66	41.75	39.97	38.68	37.63	36.39
10.jpg	49.67	46.23	44.44	42.95	41.10	38.14
11.jpg	42.38	39.12	37.31	35.98	35.10	34.17
12.jpg	41.46	38.07	35.78	34.01	32.94	31.85
13.jpg	49.67	46.55	44.74	43.43	42.54	41.60
14.jpg	39.92	36.20	34.18	32.89	32.08	31.18
15.jpg	41.96	38.98	37.38	36.20	35.36	34.19
16.jpg	44.98	40.35	38.24	37.18	36.79	36.41
17.jpg	48.01	44.62	42.31	40.52	39.21	37.70
18.jpg	40.21	38.53	37.23	36.22	35.41	34.53
19.jpg	45.19	41.98	39.95	38.20	36.98	35.36
20.jpg	48.88	45.64	43.83	42.40	41.32	40.23

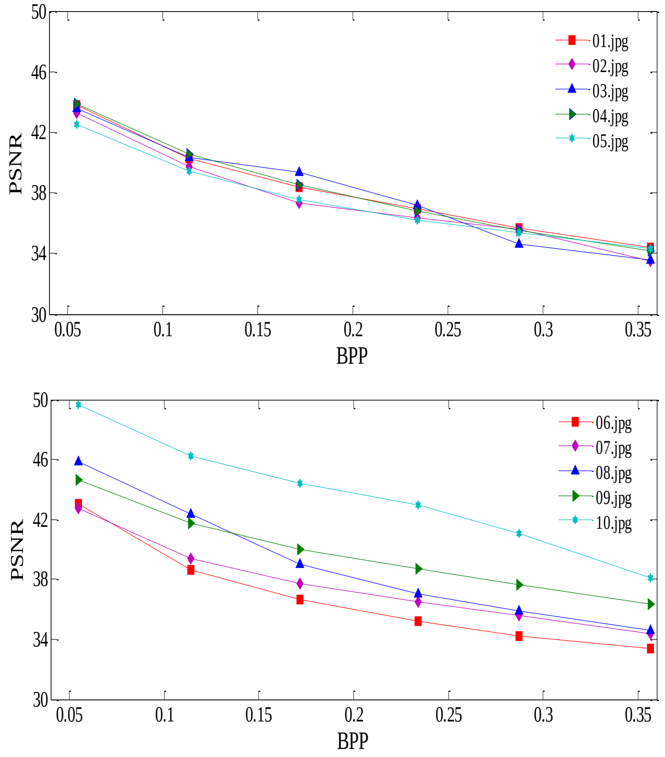
Table 2 shows the embedding capacity of proposed approach against classical approach using difference expansion. We compared our method with the Tian's approach [33] because it is a benchmark and widely adopted for comparison purposes. Moreover this is method which applies the local differences into consideration for data embedding and we expanded this method using multiple types of embedding blocks. From this table it is clear that the embedding capacity of proposed approach is higher than the classical approach for all the images used for test purpose. The hiding capacity is highest for 13.jpg and lowest for 14.jpg using proposed approach because 13.jpg has large smoother area and 14.jpg has large fine edges. It should be noted that by embedding the data bits the stego image is not same as the original image and some degradation can be seen using the PSNR in Figure 7 and Table 1 but the amount of degradation is less such that it can't be observed visually (see the input and stego images of Figure 6). The amount of degradation is less using our approach because we have not embedded any bit in blocks having more details but we embedded more number of bits in blocks having smooth details. Moreover, if the number of blocks having less details in the image are more then the amount of data using our approach will be more without loss of visual effect (see the number of bits embedded for each images in Table 2). Our experimental results indicate that the proposed solution significantly support the data hiding problem as well as it has higher hiding capacity than earlier approaches. Our algorithm embeds the amount of data according to the details of the image. More data can be attached with less detailed areas of the image (i.e. smoother part) and less data can be attached with the fine detailed areas.

TABLE II.
COMPARISON OF HIDING CAPACITY OF IMAGES BY PROPOSED AND CLASSICAL APPROACH BASED ON THE DIFFERENCE EXPANSION [33]

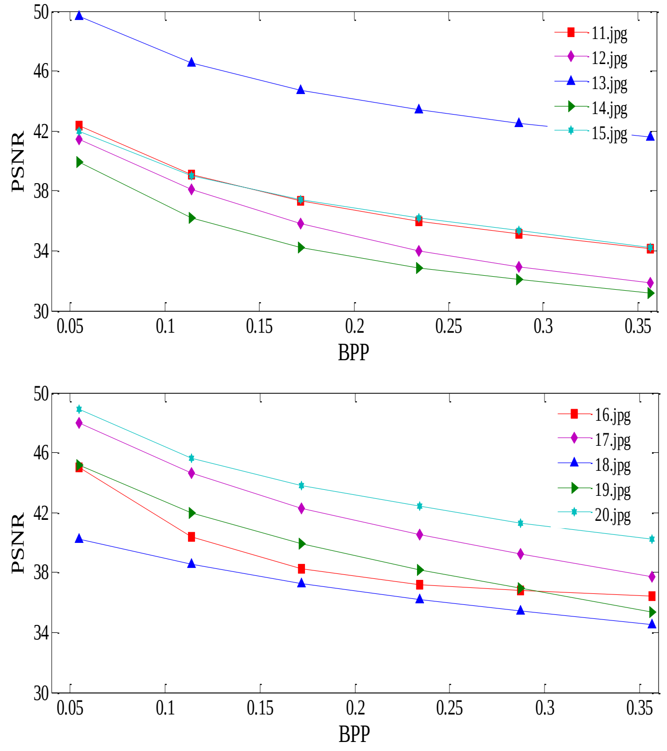
Image	**Capacity by Difference Expansion(in bits)**	**Capacity by Proposed Approach(in bits)**
01.jpg	121421	224966
02.jpg	125815	247795
03.jpg	104685	192327
04.jpg	135254	258903
05.jpg	137055	235224
06.jpg	101839	174839
07.jpg	122550	234142
08.jpg	124331	229485
09.jpg	131545	248200
10.jpg	92291	179724
11.jpg	132201	227796
12.jpg	128108	227957
13.jpg	105920	259619
14.jpg	139650	165627
15.jpg	130403	241323
16.jpg	99237	193444
17.jpg	119110	232369
18.jpg	127461	238708
19.jpg	130935	240258
20.jpg	114030	212238

V. CONCLUSIONS AND FUTURE DIRECTIONS

An image processing based approach is proposed and experimented in this paper for information protection. The proposed approach is comprised of mainly three steps. In the first step, message generation and image selection is performed using the data that is to be embedded. In the second step, we embed the data into the corresponding image. First, image is partitioned into the different number of non-overlapping blocks and then we embedded different number of bits in different blocks of the image according to the types of the block. Three types of blocks are considered in this paper according to the intensity of details of that block i.e. smooth, average and dense and embedded three, one and none bits respectively. In the third step, we extract the data and restore the image to its initial state. An approach similar to embedding one is also employed here to know the type of block. According to the type of block the bits are extracted. We have used the difference expansion technique to embed the data into the image and to extract the data from the image. The major advantage of using difference expansion technique is that now a large amount of data can be embedded into the image and there is no visible effect on the image, moreover the image can be restored to its initial state, thus our approach shows that no noise is added to the image except the data which is extracted in the extraction phase. Our experimental results indicate that the proposed solution can significantly support the data hiding problem. The future work includes the consideration of RGB color images and videos for information hiding.



(a)
(b)



(c)
(d)

Fig. 7. Graph between PSNR and BPP for (a) images 1-5, (b) images 6-10, (c) images 11-15, and (d) images 16-20

References

[1]J.C. Judge, "*Steganography: past, present, future*," SANS Institute publication, 2001.

[2]S. R. Dubey and A. S. Jalal, "Adapted Approach for Fruit Disease Identification using Images," *International Journal of Computer Vision and Image Processing*, Vol. 2, No. 3, pp. 44-58, 2012.

[3]V.B. Semwal, K.S. Kumar, V.B. Semwal and M. Sati, "Accurate location estimation of moving object with energy constraint & adaptive update algorithms to save data," arXiv preprint arXiv:1108.1321, 2011.

[4]W.L. Romero, R.G. Crespo and A.C. Sanz, "A prototype for linear features generalization," *International Journal of Interactive Multimedia & Artificial Intelligence*, Vol. 1, No. 3 2010.

[5]H. Bolivar, A. Pacheco and R.G. Crespo, "Semantics of immersive web through its architectural structure and graphic primitives," *International Journal of Interactive Multimedia & Artificial Intelligence*, Vol. 1, No. 3, 2010.

[6]S.J.B. Castro, R.G. Crespo and V.H.M. García, "Patterns of Software Development Process," *International Journal of Interactive Multimedia & Artificial Intelligence*, Vol. 1, No. 4, 2011.

[7]K.K. Susheel, V.B. Semwal and R.C. Tripathi, "Real time face recognition using adaboost improved fast PCA algorithm," arXiv preprint arXiv:1108.1353, 2011.

[8]S. R. Dubey and A. S. Jalal, "Robust Approach for Fruit and Vegetable Classification," *Procedia Engineering*, Vol. 38, pp. 3449-3453, 2012.

[9]S. R. Dubey, P. Dixit, N. Singh, and J.P. Gupta, "Infected fruit part detection using K-means clustering segmentation technique," *International Journal of Artificial Intelligence and Interactive Multimedia*. Vol. 2, No. 2, pp. 65-72, 2013.

[10] X. Li, W. Zhang, X. Gui and B. Yang, "A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification," *IEEE Transactions on Information Forensics and Security*, Vol. 8, No. 7, pp. 1091 – 1100, 2013.

[11] O.S. Faragallah, "Quadruple Difference Expansion-Based Reversible Data Hiding Method for Digital Images," *Information Security Journal: A Global Perspective*, Vol. 21, No. 5, pp. 285-295, 2012.

[12] S. R. Dubey, N. Singh, J.P. Gupta and P. Dixit, "Defect Segmentation of Fruits using K-means Clustering Technique," In *the Proceedings of the Third International Conference on Technical and Managerial Innovation in Computing and Communications in Industry and Academia*. 2012.

[13] N. Singh, S. R. Dubey, P. Dixit and J.P. Gupta, "Semantic Image Retrieval Using Multiple," In *proceeding of the 3rd International Conference on Technical and Managerial Innovation in Computing and Communications in Industry and Academia*. 2012.

[14] S. R. Dubey and A. S. Jalal, "Detection and Classification of Apple Fruit Diseases Using Complete Local Binary Patterns," In *proceeding of the Third International Conference on Computer and Communication Technology*, pp. 346-351, 2012.

[15] S. R. Dubey and A. S. Jalal, “Species and variety detection of fruits and vegetables from images,” *International Journal of Applied Pattern Recognition*. Vol. 1, No. 1, pp. 108-126, 2013.

[16] N. Singh, S. R. Dubey, P. Dixit and J.P. Gupta, “Semantic Image Retrieval by Combining Color, Texture and Shape Features,” In *the Proceedings of the International Conference on Computing Sciences*, pp. 116-120, 2012.

[17] J. P. Gupta, N. Singh, P. Dixit, V. B. Semwal and S. R. Dubey, “Human Activity Recognition using Gait Pattern,” *International Journal of Computer Vision and Image Processing*, Vol. 3, No. 3, pp. 31 – 53, 2013.

[18] K.S. Kumar, V.B. Semwal, S. Prasad and R.C. Tripathi, “Generating 3D Model Using 2D Images of an Object,” *International Journal of Engineering Science*, 2011.

[19] V.B. Semwal, V.B. Semwal, M. Sati and S. Verma, “Accurate location estimation of moving object in Wireless Sensor network,” *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 1, No. 4 , pp. 71-75, 2011.

[20] K.S. Kumar, S. Prasad, S. Banwral and V.B. Semwal, “Sports Video Summarization using Priority Curve Algorithm,” *International Journal on Computer Science & Engineering*, 2010.

[21] R.G. Crespo, S.R. Aguilar, R.F. Escobar and N. Torres, “Dynamic, ecological, accessible and 3D Virtual Worlds-based Libraries using OpenSim and Sloodle along with mobile location and NFC for checking in,” *International Journal of Interactive Multimedia & Artificial Intelligence*, Vol. 1, No. 7, 2012.

[22] C.J. Broncano, C. Pinilla, R.G. Crespo and A. Castillo, “Relative Radiometric Normalization of Multitemporal images,” *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 1, No. 3, 2010.

[23] N.F. Johnson and S. Jajodia, “Exploring steganography: seeing the unseen,” *IEEE Computer*, Vol. 31, No. 2, pp. 26–34, 1998.

[24] N. Provos and P. Honeyman, “Hide and seek: an introduction to steganography,” *IEEE Security and Privacy*, Vol. 1, No. 3, pp. 32–44, 2003.

[25] P. Moulin and R. Koetter, “Data-hiding codes,” *Proceedings of the IEEE*, Vol. 93, No. 12, pp. 2083–2126, 2005.

[26] S.B. Sadkhan, “Cryptography: current status and future trends,” In *the Proceedings of the IEEE International Conference on Information & Communication Technologies: From Theory to Applications*, pp. 417–418, 2004.

[27] C. Kurak and J. McHugh, “A cautionary note on image downgrading,” In *the Proceedings of the 8th IEEE Conference on Computer Security Applications*, pp. 153–159, 1992.

[28] C. Hosmer, “Discovering hidden evidence,” *Journal of Digital Forensic Practice*, Vol. 1, No. 1, pp. 47–56, 2006.

[29] J.C.H. Castro, I.B. Lopez, J.M.E. Tapiador and A.R. Garnacho, “Steganography in games: a general methodology and its application to the game of Go,” *Computers and Security*, Vol. 25, No. 1, pp. 64–71, 2006.

[30] C.D. Vleeschouwer, J.F. Delaigle and B. Macq, “Circular interpretation of bijective transformations in lossless watermarking for media asset management,” *IEEE Transactions on Multimedia*, Vol. 5, No. 1, pp. 97–105, 2003.

[31] M.U. Celik, G. Sharma, A.M. Tekalp and E. Saber, “Lossless generalized-LSB data embedding,” *IEEE Transactions on Image Processing*, Vol. 14, No. 2, pp. 253–266, 2005.

[32] Z. Ni, Y.Q. Shi, N. Ansari and W. Su, “Reversible data hiding,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354–362, 2006.

[33] J. Tian, “Reversible Data Embedding Using A Difference Expansion,” I*EEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, pp. 890–896, 2003.

[34] A.M. Alattar, “Reversible watermarking using the difference expansion of a generalized integer transform,” *IEEE Transactions on Image Processing*, Vol. 13, No. 8, pp. 1147–1156, 2004.

[35] S.S. Maniccam and N.G. Bourbakis, “Lossless image compression and encryption using SCAN,” *Pattern Recognition*, Vol. 34, No. 6, pp. 1229–1245, 2001.

[36] L.D. Paulson, “New system fights steganography, News Briefs,” *IEEE Computer Society*, Vol. 39, No.8, pp. 25–27, 2006.

[37] A.A. Abdelwahab and L.A. Hassaan, “A discrete wavelet transform based technique for image data hiding,” In *the Proceedings of the 25th National Radio Science Conference*, pp. 1–9, 2008.

[38] V.M. Potdar, S. Han and E. Chang, “Fingerprinted secret sharing steganography for robustness against image cropping attacks,” In *the Proceedings of the IEEE Third International Conference on Industrial Informatics*, pp. 717–724, 2005.

[39] C.C. Lin, W.L. Tai and C.C. Chang, “Multilevel reversible data hiding based on histogram modification of difference images,” *Pattern Recognition*, Vol. 41, No. 12, pp. 3582–3591, 2008.

[40] Y.T. Wu and F.Y. Shih, “Genetic algorithm based methodology for breaking the steganalytic systems,” *IEEE Transactions on Systems, Man, and Cybernetics—part B: cybernetics*, Vol. 36, No. 1, pp. 24–31, 2006.

[41] K.B. Raja, S. Sindhu, T.D. Mahalakshmi, S. Akshatha, B.K. Nithin, M. Sarvajith, K.R. Venugopal and L.M. Patnaik, “Robust image adaptive steganography using integer wavelets,” In *the Proceedings of the Third International Conference on Communication System Software and Middleware*, Bangalore, India, pp. 614–621, 2008.

[42] S.R. Dubey, “*Automatic Recognition of Fruits and Vegetables and Detection of Fruit Diseases*”, M.Tech Thesis, GLAU Mathura, India, 2012.

[43] M. Sati, V. Vikash, V. Bijalwan, P. Kumari, M. Raj, M. Balodhi, P. Gairola and V.B. Semwal, “A fault-tolerant mobile computing model based on scalable replica”, *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. 2, 2014.

[44] V.B. Semwal, S.A. Katiyar, P. Chakraborty and G.C. Nandi, “Biped Model Based on Human Gait Pattern Parameters for Sagittal Plane Movement”, *CARE*, 2013.

[45] V.B. Semwal, A. Bhushan and G.C. Nandi, “Study of Humanoid Push Recovery Based on Experiments”, *CARE*, 2013.

[46] S.R. Dubey and A.S. Jalal, “Automatic Fruit Disease Classification using Images”, *Computer Vision and Image Processing in Intelligent Systems and Multimedia Technologies*, 2014.

[47] D.D. Agrawal, S.R. Dubey and A.S. Jalal, “Emotion Recognition from Facial Expressions based on Multi-level Classification”, *International Journal of Computational Vision and Robotics*, 2014.

[48] S.R. Dubey and A.S. Jalal, “Fruit Disease Recognition using Improved Sum and Difference Histogram from Images”, International Journal of Applied Pattern Recognition, 2014.

Pushkar Dixit is currently working as an Assistant Professor in Computer Science and Engineering, FET Agra College, Agra, India. He received his MTech in Computer Science from GLA University Mathura, India and BTech in Computer Science from FET Agra College Agra, India. He has 3 year of teaching and research. His research interests include image processing, network security and computer networks.

Nishant Singh received his MTech in CSE in 2012 from GLA University Mathura, India and BTech in CSE in 2010 from B.S.A. College of Engineering and Technology Mathura, India. He has 1 year of teaching and research experience and currently, he is working as an Assistant Professor in Department of Computer Engineering and Applications, Poornima Institute of Engineering and Technology, Jaipur, India. His research interests include image processing, computer vision and pattern recognition.

Jay Prakash Gupta received his MTech in CSE in 2012 from GLA University Mathura, India and BTech in CSE in 2010 from B.S.A. College of Engineering and Technology Mathura, India. He is working as a System Engineer in Infosys Limited, Pune, India. His research interests include image processing, computer vision and pattern recognition.