WATERMARKING ALGORITHM TO DETECT IMAGE TAMPERING USING IMPROVED ADAPTIVE HUFFMAN CODING TECHNIQUE

Vinoth.V.V
Research Scholar
Bharath University
Chennai
vinfo.vv@gmail.com

Dr.T.Krishnakumar
Head- Strategy and Planning
Bharath University
Chennai
drkk@bharathuniv.ac.in

***Abstract*— Watermarking algorithms have been widely applied to the original images to protect them against tampering has recently attracted an overgrowing interest. In the existing approaches SPIHT algorithm is used to compress the images of fragile watermarks are commonly used for tamper detection, authentication and content integrity verification. But in fragile watermarking extraction process was going to be fail if attack was happened on digital signal is attacked. So to overcome this problem by introducing the modified fragile watermarking technique for image recovery. For this purpose, we need to design a watermarking algorithm fulfilling two purposes in case of image tampering: 1) detecting the tampered area of the received image and 2) recovering the lost information in the tampered zones. In this paper the watermarking using DCT technique is been combined with image compression technique using improved adaptive Huffman encoding.In watermark embedding phase, the original image is source coded and the output bit stream is protected using appropriate channel encoder. For image recovery, erasure locations detectedby check bits help channel erasure decoder to retrieve theoriginal source encoded image. The improved adaptive Huffman coding technique is been based on Huffman algorithm. This new compression algorithm not only reduces the number of times required to encode the data but at the same time reduces the space for storing in comparison to adaptive Huffman and static Huffman respectively.The watermarked image quality gain is achieved through spending less bit-budget on watermark, while image recoveryquality is considerably improved as a consequence of consistent performance of designed source and channel codes.**

Index Terms:- SPIHT, Image tampering protection, Discrete Cosine Transform

I. INTRODUCTION

Watermarking can be defined as process of hiding the digital information. In watermarking carrier signal is generated and digital information is going to hide in carrier signal with a special care that there is a no relation between carrier signal and digital information. In entire process of water marking has three major steps are include like embedding, attack and detection. In the process of embedding with the help of watermark signal and host signal i.e. original data we always used to generate watermark signal. Then over the transmission media watermark signal transmitted to receiver from sender. During transmission process suppose any changes or modifications are done in transmitted signal then it is called attack. Then watermark signal is automatically altered while detection is the process of extracting the watermark from obtained signal whether the signal was attacked or not. If signal was not modified then watermark will present as it is and can be extracted easily.A watermarking has two basic types includes robust watermarking and fragile watermarking. For copyright and ownership authentication we use robust watermarking while to check integrity and authenticity of digital images we use fragile watermarking [11]. We can detect any change in the digital images or signals. In robust watermarking theory the attack is very strong in digital image, with the help of extraction algorithm we can produce the watermark accurately.But in fragile watermarking extraction process was going to be fail if attack was happened on digital signal is attacked.In the existing approaches a robust invisible watermarking algorithm is using discrete wavelet transform based edge detection. Image is decomposed to sub-band coefficients using the wavelet transform. The shift invariant edge detection is applied to high frequency sub-band to find the edge coefficients. The watermark is embedded within the selected sub-band coefficients near the edges. Morphological dilation is used along with the edge coefficients for improving the robustness of the watermarking. As adding the watermark in high frequency sub-bands it will degrade the invisibility thus, scaled dilated edge coefficients are used to

improve the invisibility. Performance of the method is tested on the different images and evaluated based on MSE, PSNR and NCC. It is found that the method improves the invisibility of the watermark and is robust to various attacks such as compression, cropping and resizing.

In the Proposed approaches has been presented to provide security at enhanced level. Here the two techniques namely watermarking and Tampered detection are combined together to enhance the level of security for data transmission purpose. Watermarking using DCT technique is been combined with image compression technique using improved adaptive Huffman encoding. The improved adaptive Huffman coding technique is been based on Huffman algorithm. This new compression algorithm not only reduces the number of pass required to encode the data but at the same time reduces the storage space in comparison to adaptive Huffman and static Huffman respectively. Therefore the method completely extracts the embedded data from watermarked images. We investigated the performance of the proposed method in view of the image quality and the ability of tamper detection and recovery. The experiments of the image quality confirmed the imperceptibility of watermarked images and images used for recovery. The experiments of the ability of tamper detection and recovery confirmed the maximum size of recoverable tampered region and the difficulty of undetectable falsification.The drawback of existing approach is Interpolation-based approaches tend to blur high frequency details if the up-scaling ratio is large and if the low-resolution image is generated with anti-aliasing operation. The computational complexity of learning-based super-resolution approaches is quite high.

In the existing methods there are two drawbacks, namely

1.Watermarking in spatial domain

2.Embedding robustness

Watermarking in spatial domain

Hiding process of digital information in an image is known as watermarking. This type of hidden process does not contain any relation to an image. Watermarking is simply a technique used to verify the owner's data and authentication from the nature and non-nature image to ensure the symbol of owner-ship. Using this method, we verify the owner data i.e., signature, facsimile, etc. This identifies the original owner by extracting and detecting from the watermarked images. Owner can ensure the multimedia data which belongs to them using embedding the watermark into the original image. The human eye cannot differentiate the watermark from the watermarking image. Based on the domains used in the embedded watermarks, the watermarking is classified into transform and spatial water marking. Higher data embedding applications uses spatial domain watermarking. The transform domain technique is suitable for the application where the robustness is a crucial concern. The basic block diagram of the watermarking is shown in fig 1.

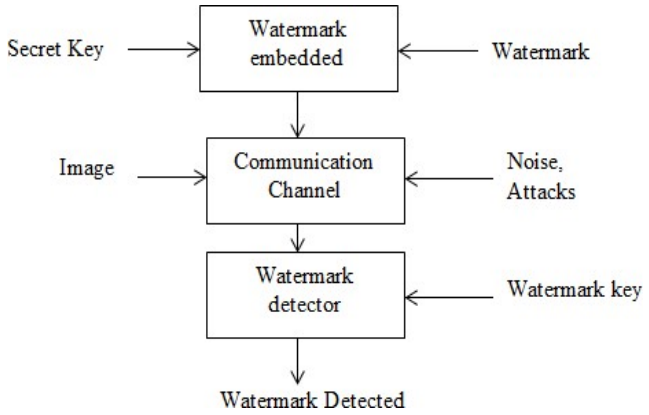

Fig.1 Watermarking process

In previous days, the watermarking comprises a sequence of bits and it can be detected using detection theory. On the verification time, the original image is subtracted from the given image, verified and identified. Self-recovery methods for watermarked bits is of two categories: check bits and reference bits, check bits are used to locate the tampered blocks in the image, the reference bits used to restore the original image. In digital watermarking, the main advantage is image authentication. Digital watermarking uses a secret key and the key is used at the decoder to decode the watermarked image. The characteristics of Digital watermarking technique are security, imperceptibility and capacity. The main features of watermarking technique are,

Noiselessness: It is not noticed by simple analyzing.

Key distinctiveness: watermarking keys are statistically independent.

Robustness: Using common signal processing operations, it is efficiently detected.

The spatial domain is used in watermarking method. A common picture cropping operation that may be used to eliminate the watermark is the major disadvantage in spatial domain watermarking. So frequency domain approaches have also been proposed. Multimedia watermarking is also used spread spectrum technique.

Embedding Robustness:

Robustness is defined as recover the watermark from the watermarking images for after several attacks takes place. For example attacks are cropping, rotation, Gaussian etc. The important issue that needs to be concerned in watermarking is security and robustness. The facility to restrict the embedding, extracting and un-authorized removal from an image is known as security.

II. TECHNIQUES OF DIGITAL WATERMARKING

1. Spatial Domain Method

Spatial-domain method is used for embedding the watermarks into a particular text, image by directly changing the pixel values of original host images. Some common spatial-domain algorithms include Least Significant Bit (LSB) Modification, Patchwork, Texture Block Coding, etc. The most serious drawback of spatial-domain technologies is that it tends to provides limited robustness. It is complex for spatial-domain watermarks to subsist under attacks such as lossy compression and low-pass filtering. Also the amount of information that can be embedded in spatial domain is also very limited.

2. Frequency-Domain Technologies

In comparison to spatial-domain watermark, watermarks in frequency domain are more robust and much more compatible to popular image compression standards. Thus frequency-domain watermarking technique is more widely used and obtains more attention in comparison to spatial domain method. To embed a watermark, a frequency transformation needs to be applied to the host data. Then, modifications are made to the transform coefficients. Possible frequency image transformations include the Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and others. In recent years they are becoming generally desolated.

III. APPLICATIONS OF WATERMARKING

1. Copyright Protection

This is one of the most prominent applications of watermarks. Due to huge exchange of images over insecure networks, copyright protection becomes a very important issue. Watermarking an image will prevent its redistribution.

2. Authentication

In some cases there arises the need to identify the ownership of the contents. All this can be done by embedding a watermark and providing the owner with a private key that gives him an access to the message. ID cards, ATM cards, credit cards are all examples of documents that require authentication.

3. Broadcast Monitoring

From the name it is clear that broadcast monitoring is been used to verify the programs that are broadcasted on TV or radio. It especially helps the advertising companies to see if their advertisements appeared for the right duration or not.

4. Content Labelling

Watermarks can be used for providing more information about the cover object. This process is named content labelling.

5. Tamper Detection

Fragile watermarks can be used to detect tampering in an image. If the fragile watermark is degraded in any way then we can say that the image or document in question has been tampered [5].

6. Digital Fingerprinting

This is a process that is been used for detecting the owner of the content. This is so because every fingerprint is the unique characteristics of the owner.

7. Content protection

In this process the content stamped with a visible watermark that is very difficult to remove so that it can be publicly and freely distributed.

IV.IMAGE COMPRESSION TECHNIQUES

The image compression techniques are broadly classified into two categories depending whether or not an exact repro of the original image could be reconstructed using the compressed image. These are:

1. Lossless technique
2. Lossy technique

1) Lossless compression

It is a compression technique that does not lose any data in the compression process. Lossless compression "packs data" into a smaller file size by using a kind of internal shorthand to signify redundant data. If an original

file is 1.5MB (megabytes), lossless compression can reduce it to about half of that size, depending on the type of file that is being compressed. This makes lossless compression convenient for transferring files across the Internet, as smaller files transfer faster [11]. Lossless compression is also handy for storing files as they take up less room. The zip convention, used in programs like WinZip, uses lossless compression. For this reason zip software is popular for compressing program and data files. That's because when these files are decompressed, all bytes must be present to ensure their integrity. If bytes are missing from a program, it won't run. If bytes are missing from a data file, it will be incomplete and falsified. GIF image files also use lossless compression. Lossless compression has advantages as well as disadvantages. The advantage is that the compressed file will decompress to an exact duplicate of the original file, mirroring its quality. The disadvantage is that the compression ratio is not all that high, precisely because no data is lost. Following techniques are included in lossless compression:

1. Run length encoding
2. Huffman encoding
3. LZW coding
4. Area coding

2) Lossy Compression

It is a compression technique that does not decompress data back to 100% of the original. Lossy methods provide high degrees of compression and result in very small compressed files, but there is a certain amount of loss when they are restored [13]. Audio, video and some imaging applications can tolerate loss, and in many cases, it may not be noticeable to the human ear or eye. In other cases, it may be noticeable, but not that critical to the application. The more tolerance for loss, the smaller the file can be compressed, and the faster the file can be transmitted over a network. Examples of lossy file formats are MP3, AAC, MPEG and JPEG. Lossy compression is never used for business data and text, which demand a perfect "lossless" restoration. Lossy schemes tend to provide much higher compression ratios than lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate for most applications. By this scheme, the decompressed image is not identical to the original image, but reasonably close to it. Lossy compression techniques includes following schemes:

1. Transformation coding
2. Vector quantization
3. Fractal coding
4. Block Truncation Coding
5. Sub band coding

V.PROPOSED SYSTEM

In the Proposed System, watermarking can alone make the data transmission completely secure. Hence the new technique is been proposed in order to achieve secure transmission of data by making combine use of image watermarking using DCT technique and then applying image compression technique using improved adaptive Huffman algorithm to it. The DCT allows an image to be split up into the different frequency bands, making it convents for embedding the watermarking information into the middle frequency bands of an image. The middle frequency bands are chosen such that they have minimized they avoid the most visual important parts of the image (low frequency) without over-exposing themselves to removal through compression and noise attacks. Perform DCT transformation on watermarked image and the original host image. After doing so, subtract original host image from watermarked image. And finally do multiplication of extracted watermark by scaling factor to display.

After image has been watermarked it is been compressed using a compression algorithm that is based on Huffman coding. Huffman coding is one of the lossless compression techniques. Huffman algorithms have two ranges static as well as adaptive. Static Huffman algorithm is a technique that encodes the data in two passes. In first pass the frequency of each symbol is been calculated and in second pass a Huffman tree needs to be constructed. Adaptive Huffman algorithm is extended on the basis of Huffman algorithm that constructs the Huffman tree in one pass but take more space in comparison to Static Huffman algorithm. The algorithm that is used not only reduces the number of pass but also reduce the storage space needed in comparison to adaptive Huffman algorithm and comparable to static. We investigated the performance of the proposed method in view of the image quality and the ability of tamper detection and recovery. The experiments of the image quality confirmed the imperceptibility of watermarked images and

images used for recovery. The experiments of the ability of tamper detection and recovery confirmed the maximum size of recoverable tampered region and the difficulty of undetectable falsification.

LSB Detection

Consider an image represented by 8 bit gray scale pixel values. These 8 bits are divided into Most Significant Bits; Check bits (n_c), Source bits (ns) and Channel code parity/redundancy bits (n_p). The Check bits (n_c), Source bits (n_s) and Channel code parity/redundancy bits (n_p) will form the LSB. The MSB will not be changed at the time of embedding a watermark and later used for image recovery. Since this LSB bits are replaced with watermark bits, it is detected using a LSB detection block and its mean square error is given by

$$\text{MSE}(n_{w}) = \frac{1}{2^{nw}} \sum_{i=0}^{2^{nw}-1} \sum_{j=0}^{2^{nw}-1} (i-j)^2 = \frac{4^{nw}-1}{6} \qquad (1)$$

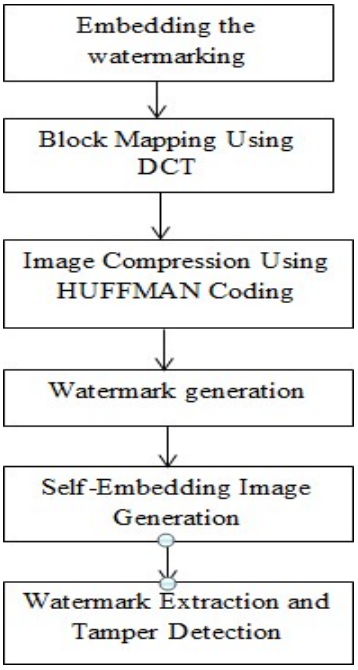

Fig 2. Flow diagram of proposed method

Embedding the Watermarking

In this process of embedding of watermark we hide watermark information into LSB (Least significant Bit) of the next randomly generated block in the block–chain to get watermark image. There are so many watermarking algorithms used based on the type of application. A general watermark embedding process is shown in fig 2. The bits other than Most Significant Bits (i.e., check bits and channel encoded bits) are used to embed a watermark in the original image. As said in earlier, check bits and channel coded bits generate a secret key which is used to protect it from tampering and secured authentication. It finds application in forensic imagery and defence.

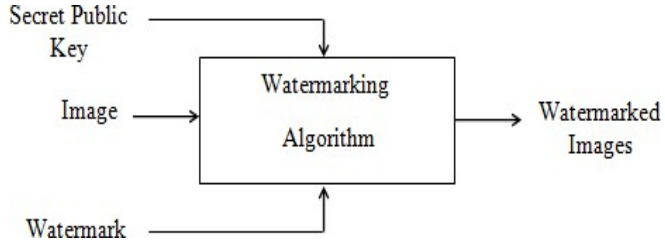

Fig 3. Embedding Process

Consider the original image *I* represented by 8-bit gray-scale pixel values. These eight bits are divided into four parts: The most significant bits that will not change at the watermark embedding phase, check bits, source code bits, and channel code parity bits, denoted by n_m, n_h, n_s and n_p respectively. The n_m MSB of each pixel is remained unchanged during watermark embedding and will be used later for hash generation and image reconstruction. The remaining bits are used for the purpose of watermark embedding. Assume that *N* denotes the number of image pixels. We compress the original image into $Ns = N \times n_s$ bits using proper source coding algorithm. Huffman is an embedded compression algorithm, i. e., one can extract an estimation of the original image by truncating its output in every desired rate. This property which fits our design of a general framework, together with the high compression gain when applied over the whole image.

Block Mapping

As we have the original image is portioned in non-overlapping block of N having size so as we may say that, so for each block we can say that there is one watermark block is present. We can form sequence of block in randomized fashion. For generation of recovery information of each block

we need a mapped block having that block index number, recovery information, so we use DCT and inverse DCT process for this. As all blocks are to be completed with DCT will insert this information into the LSB plane of next randomly selected block in block chaining.

The DCT allows an image to be split up into the different frequency bands, making it convenient for embedding the watermarking information into the middle frequency bands of an image. The middle frequency bands are chosen such that they have minimized they avoid the most visual important parts of the image (low frequency) without over-exposing themselves to removal through compression and noise attacks.

Image compression using Huffman Coding

Consider grey level images for implementing the Huffman coding technique. Grey scale is a sequence of grey shades ranging from black to white with intermediate shades of grey. It ranges from (0, 2), n is the number of bits. Here we consider grey level values of 8- bit grey scale image range from 0 to 255. An image is a m*n matrix. In Huffman coding, the first is to find the frequency of occurrences of each grey value. Now place the grey values in the descending order of the frequencies. Thus the grey value with maximum frequency gets the code word with minimum length and the grey value with minimum frequency gets the code word with maximum length. Now the grey values in the image are replaced with the respective code words so as to compress the image and then the compressed secret image is embedded into the original image with the help of modified auxiliary carry method.

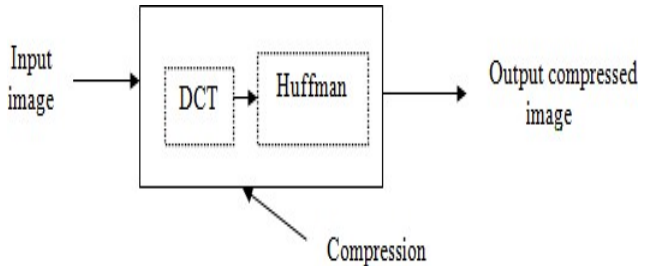

Fig. 4 Schematic diagram for the image compression algorithm using DCT Huffman encoding.

Secret data it will be encoded by Huffman coding. Huffman coding is an entropy encoding algorithm used for lossless data compression developed by David A. Huffman. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It can be informally defined as a prefix-free binary code (a set of code words) with minimum expected codeword length (equivalently, a tree with minimum weighted path length). Formally it can be defined as follow : Input: (i) Let the input be an array of alphabet A= {a1, a2, …..an},which is the symbol alphabet of size n. (ii) Let W= {w1,w2,……wn}, which is the set of the positive symbol weights Wi=weight(ai), 1≤i≤n Output: Code C(A,W) = {C1,C2,…..Cn}, which is the set of (binary) code words, where Ci is the codeword for ai, 1≤i≤n. To illustrate the work of Huffman coding assume the string "go go gophers" is encoded in ASCII, how we might save bits using a simpler coding scheme, and how Huffman coding is used to compress the data resulting in still more savings.

Algorithm

1. Scan the first Symbol and initialize its frequency to 1

2. Then next symbol is been scanned from the source data If any previous symbol = next symbol then the frequency of that previous symbol needs to be incremented If any previous symbol frequency < recently incremented symbol frequency then both nodes needs to be interchanged Else Initialize their frequency to 1

3. Create strictly binary tree with left and right node (Left or Right node can be NULL). The root is the composite Symbols of left and right nodes. Assign value 0 to Right node and 1to Left node.

4. Step 2 to 4 needs to be repeated till End of Source data is been reached. By the use of this algorithm the storage space will be reduced and time is also been saved. Thus the combination of the two processes watermarking and compression will results in providing high security level to the data to be transmitted

Advantages:

- Improved adaptive Huffman utilizes less space to store the compressed data.

- It saves the time because here, there is no need to scan the whole string for constructing the first tree. It also saves the time while constructing trees e.g. it needs only one symbol for constructing the first tree unlike in adaptive Huffman that requires all different symbols to construct the tree.
- In Improved adaptive Huffman even if one symbol occurs frequently will tend to have same code.
- Finally during the process of decompressing the only final tree is needed.
    - It gives promising result to image recovery from tampered area effectively.
    - It gives successful response in tampering detection and image recovery of color images.

Watermark generation

We will generate watermark and embed it into LSB of next randomly generated block in block chaining. For respective block which is having the entire information block as it but except LSB. As LSB is set to zero, whatever watermark information is generated that we embedded into LSB of next randomly generated block in block chaining and finally we get watermarked of image block.

Self-Embedding Image Generation

The original image is portioned in non-overlapping, so for each block there is one self-embedded block is presented. We can form sequence of block in randomized fashion and again we have to note pairing. For generation of recovery information of each block we need a mapped block having that block index number, recovery information. As all blocks are to be completed with DCT, we will insert this information into the LSB plane of next randomly selected block in block chaining and will generate self-embed image information . The entire above are repeated for each block of original image to generate self-embedded image.

Huffman Decompression

Generally speaking, the process of decompression is simply a matter of translating the stream of prefix codes to individual byte values; usually by traversing the Huffman tree node by node as each bit is read from the input stream). Before this can take place, however, the Huffman tree must be somehow reconstructed. In the simplest case, where character frequencies are fairly predictable, the tree can be preconstructed (and even statistically adjusted on each compression cycle) and thus reused every time, at the expense of at least some measure of compression efficiency. Otherwise, the information to reconstruct the tree must be sent a priori. In any case, since the compressed data can include unused "trailing bits" the decompressor must be able to determine when to stop producing output. This can be accomplished by either transmitting the length of the decompressed data along with the compression model or by defining a special code symbol to signify the end of input

Watermark Extraction and Tamper Detection:

In watermark extraction procedure, we have to follow reverse procedure of watermark embedding. The LSB plane of watermark block is set to zero and generates its complement block. Then we will calculate hash function and then we will perform pixel to pixel EXOR operation along with the LSB plane of next block which presents in block chaining mapping sequence. We use this block chaining efficiently to produce recovery information with secret key K for extraction of watermark.

The received image which is probably tampered is decomposed into blocks of size B×B. For each block, position bits are found using k_2, derived from shared secret key. Block bits are decomposed to nm MSB and n_w watermark LSB per pixel (bpp), which results in $b_m = n_m \times B_2$ MSB and $b_w = n_w \times B_2$ watermark bits. The watermark bit stream itself is decomposed into $b_h = n_h \times B_2$ check bits and $b_c = n_c \times B_2$ channel code bits. b_{rc} position bits along with bm MSB are used to generate b_h hash bits. The XOR of calculated hash bits and extracted check bits is recorded for each block. For unaltered blocks, this bit stream equals the random key used in the embedding phase. Therefore, comparing these results and spotting the different ones lead to locating the tampered blocks. The probability of missing a tampered block equals 2^{-bh}, which is almost zero for sufficiently large b_h. After locating the tampered blocks, the N_c channel code bits are collected through the whole image. Channel code bits are undergoing proper inverse permutation. Then, they are delivered as input to decoder along with the erasure locations calculated from the list of tampered blocks. The compressed image bit stream available at the output of the decoder is passed through the source decoder after undergo- ing proper inverse permutation. The output of source decoder is the reconstructed image.

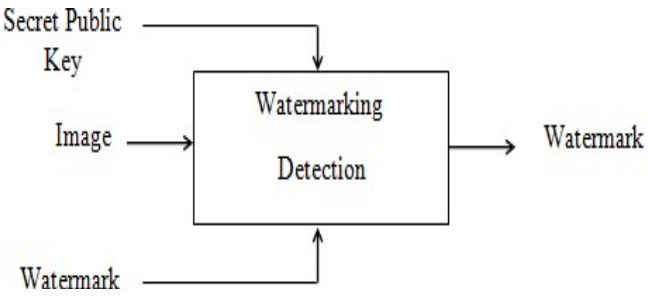

Fig 5. Decoding Process

VI.RESULTS AND DISCUSSION

The original Lena image is shown in Fig 6. Now the grey values in the original image are replaced with the respective code words (Huffman coding), so as to compress the image and then the compressed secret image is embedded into the original image with watermark text. Fig. 8 shows the watermarked image generated by the algorithm.Fig.9 shows the extracted watermarked image at the output. In the decoding process the inserted watermark text is also extracted.

Fig 6. Input Image

Fig 7. Compressed Image

The image with tampering is shown in Fig 10. In order to protect the image against high-rate tampering; we need to spend more bit-budget for watermark embedding. Fig. 10 shows the tampering image.Tampered blocks are recognized and their information is perfectly recovered as in Fig.11.

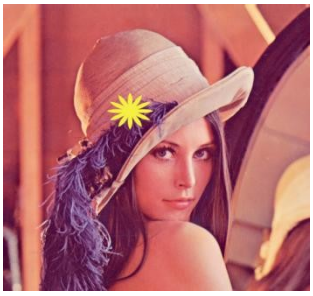

Fig 10. Image With Tampering

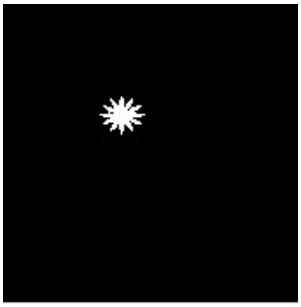

Fig 11. Tampering Detection

VII. CONCLUSION

Watermarking scheme is used to protect the original image from damaging/tampering. Then the LSB bits are divided into source encoder bits, check bits and channel encoder bits.. A tampering model is modelled to find the erasure error. This error is utilized by the channel decoder in recovering the original image. In this paper, the implementation of encoder and decoder circuits using Huffman algorithm. A better image recovery is achieved using these techniques. As a result, the quality of the restored image is high when the tampering rate is low, and the tampered image can be recovered even in the very high tampering rates. In the Proposed approaches has been presented to provide security at enhanced level. Here the two techniques namely watermarking and Tampered detection are combined together to enhance the level of security for data transmission purpose.

REFERENCES

[1] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230,Jun. 2006.

[2]] S. Roy and Q. Sun, "Robust hash for detecting and localizing image tampering," in Proc. *IEEE Int. Conf. Image Process. (ICIP), vol. 6*. Sep./Oct. 2007, pp. VI-117–VI-120.

[3] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Trans. Image Process.*, vol. 18,

no. 11, pp. 2491–2504, Nov. 2009.

[4] M. Wu and B. Liu, "Watermarking for image authentication," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 2. 1998, pp. 437–441.

[5] J. Fridrich, "Image watermarking for tamper detection," in *Proc. Int. Conf. Image Process. (ICIP)*, vol. 2. Oct. 1998, pp. 404–408.

[6] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proc. IEEE*, vol. 87, no. 7, pp. 1167–1180, Jul. 1999.

[7] C.-S. Lu, S.-K. Huang, C.-J. Sze, and H.-Y. M. Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. Multimedia*, vol. 2, no. 4, pp. 209–224, Dec. 2000.

[8] Shih-Hsuan Yang And Wu-Jie Liao "Evaluation Of SPIHT Coding Parameters"*IEEE Transactions On Image Processing, Vol*. 24, No. 7, July 2015

[9] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001.

[10] M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 585–595, Jun. 2002.

[11] D. Zou, Y. Q. Shi, Z. Ni, and W. Su, "A semi-fragile lossless digital watermarking scheme based on integer wavelet transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1294–1300, Oct. 2006.

[12] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[13] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, and M. Carli, "A robust error concealment technique using data hiding for image and video transmission over lossy channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1394–1406, Nov. 2005.

[14] G. Gur, Y. Altug, E. Anarim, and F. Alagoz, "Image error concealment using watermarking with subbands for wireless channels," *IEEE Commun. Lett.*, vol. 11, no. 2, pp. 179–181, Feb. 2007.

[15] P. Korus and A. Dziech, "Efficient method for content reconstruction with self embedding," *Image Processing,IEEE Transactions on*,vol. 22, no.3, pp. 1134–1147, 2013.

[16] Xinpeng Zhang, Zhenxing Qian, Yanli Ren, and Guorui Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive recon- struction," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 4, pp. 1223–1232, 2011.