Privacy Protection in Medical Images Using Histogram Shifting Based RDH

[1]Anjali O, [2]Beena M V
[1]M.Tech Student, [2]Assistant Professor
CSE, VAST, Thrissur, India

***Abstract* - In the medical field the patients' confidential medical records need to be transmitted securely. One method would be to embed these confidential data in medical images and extract it at the other end. But while performing this embedding and extraction no data loss or image distortion should occur. So here this project presents a method of embedding the medical data in medical images using histogram shifting based reversible data hiding. An embedding process is used here where the difference between adjacent pixels in the image is utilized. Since external data is being embedded into the original image, some noticeable changes can occur in the image which might affect the security of the data. To avoid this, after data embedding, the output image should be identical to its original image. Also after extraction the doctor should be able to provide the proper treatment using the images and the extracted data. Reversible data hiding is a newly developed branch in data hiding or watermarking researches. Histogram shifting (HS) is a useful technique of reversible data hiding (RDH). With HS-based RDH, high capacity and low distortion can be achieved efficiently and both the original image and the hidden data can be perfectly recovered. The method is reversible in the sense that the extraction process is exactly opposite to embedding process. Finally the performance can be evaluated by mean square error, peak signal to noise ratio, structural similarity index etc. In future, to push forward the capacity-distortion behaviour of RDH, more meaningful shifting and embedding functions are expected.**

***Index Terms* - Histogram Shifting (HS), Reversible Data Hiding (RDH)**

I. INTRODUCTION

For most image data hiding methods, the host image is permanently distorted and it cannot be restored from the marked content. But in some applications such as medical image sharing [1], multimedia archive management [2], and image transcoding [3], any distortion due to data embedding is intolerable and the availability of the original image is in high demand. To this end, a solution called reversible data hiding (RDH) is proposed, in which the host image can be fully restore dafter data embedding [4]. RDH is a hybrid method which combines various techniques to ensure the reversibility. Its feasibility is mainly due to the lossless compressibility of natural images. The project presents privacy protection of medical images with information using histogram shifting based reversible data hiding. Medical data is embedded in to the medical image which is very useful for doctors. In this method first the host image is divided in to non-overlapping blocks such that each block contains n pixels. Then, an n-dimensional histogram is generated by counting the frequency of the pixel value-array sized n of each divided block. Finally, data embedding is implemented by modifying the resulting n-dimensional histogram. The pixel-value-array is an element of Z^n, which then needs to be divided into two disjointed sets, one set is used to carry hidden data by expansion embedding, and the other set is simply shifted to create vacant spaces to ensure the reversibility

II. OVERVIEW OF PROPOSED METHOD

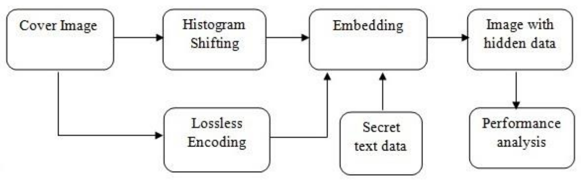


Figure 1 Architecture-Embedding

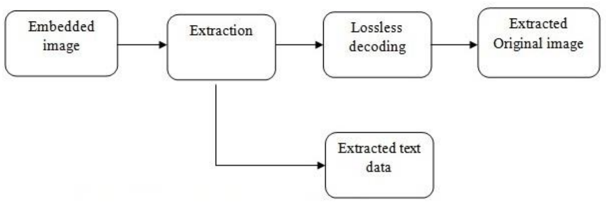


Figure 2 Architecture-Extraction

The proposed encryption and multiplexing technique not only enhances the security of confidential fingerprint information by making the information inaccessible to any interloper having a random code, but also improves the system resource utilization by reducing the storage and/or transmission bandwidth requirement.

A. Histogram Shifting

HS-based algorithm is another important work of RDH, in which the peak of image histogram is utilized to embed data. In this method, each pixel value is modified at most by 1, and thus the visual quality of marked image is guaranteed. Lee etal.s (16) proposed a method by using the histogram of difference image. This method outperforms Ni etal. s by improving both EC and visual quality. The spatial correlation of natural images is exploited in Lee etal.s method and thus a more appropriate histogram is obtained. In other words, compared with the ordinary one dimensional histogram, the difference-histogram isbetter for RDH since it is regular in shape and has a much higher peak point. Later Hong et al. (17) proposed a new HS-based method by modifying the prediction error histogram. This method can well exploit the image redundancy and thus achieve a better performance compared with the previously introduced DE based methods. Recently, Wu and Huang (4) proposed a novel HS-based method where the histogram bins used for expansion embedding are specifically selected such that the embedding distortion is minimized. The experimental results reported has demonstrated that this method is better than some state-of-the-art works including the most recently proposed integer-transform-based method. A series of experiments have been conducted to test the proposed algorithms on images. Each test image is a gray scale one of size 512X512 with the gray values ranging from 0 through 255.An algorithm is implemented for the purpose of comparing the results with those of other methods. Each test image was divided into blocks of sizes from 256X256 down to 2X2. As can be observed, as the number of blocks in each test image increases, both the PSNR of the resulting stego-image and the data hiding capacity increase until a bottleneck at the block size of 8X8 is encountered.

B. Encoding Process

Encoding is the process of converting data from one form to another. Run length encoding is used here. It is a very simple form of data compression in which runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) are as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs: for example, relatively simple graphic images such as icons, line drawings and animations. It is not useful with files that don’t have many runs as it could potentially double the file size.

C. Embedding Process

The embedding procedure contains several steps. First, after dividing the host image into non-overlapping blocks, the blocks are further divided into three parts to get I_1,I_2 and I_3. Then, by using shifting and embedding functions, embed the hidden data into $I_1$and I_3. Next, by using LSB replacement, embed the location map which records the underflow/overflow locations into I_1. Notice that, before replacing LSBs, the original LSBs of I_1 should be recorded into a LSB sequence. Finally, embed the LSB sequence intoI_2 using shifting and embedding functions. Here, the partition of three parts is to solve the underflow/ overflow problem by embedding the location map into the host image. The part I_1 is double embedded to embed first the hidden data (using shifting and embedding functions) and then the location map (using LSB replacement).The detailed Data embedding procedure is described as below in Algorithm 1. The core of this procedure is Step 4 (in algorithm 1) and the partition in Step 3 is to deal with the underflow/overflow problem. It should be mentioned that there is another commonly used way to encode the location map: in Step 2, one can get LMc by losslessly compressing LM. However, in HS-based RDH algorithms, there are usually only a few blocks which may cause underflow/overflow. We then prefer to record those problematic locations rather than compressing the location map because the latter solution is time-consuming. Lossless data hiding method based on histogram shifting embeds large volume data into cover images. It also produces stego-images with high qualities by using a strategy of hierarchical block division. The bottleneck of data-hiding rate increasing at the block size of 8X8 found in existing methods is broken by the proposed non-recursive algorithms. And the proposed recursive versions of the algorithms enhance the performance further both in the data hiding capacity and the PSNR value.

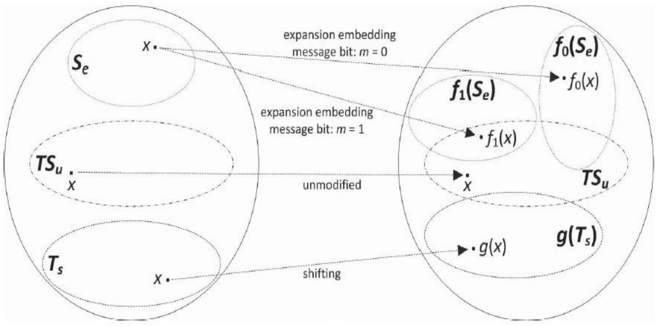


Figure 3Illustration of the data embedding

Algorithm 1 Algorithm to create embedded image

Step 1. Divide the host image into k non-overlapping blocks $\{X_1 ... X_k\}$ such that each X_i contains npixels Assume the value of X_i is $x_i \in A_n$

Step 2. Define the location map LM: LM(i) = 0 if $x_i \in T_s \cup Se$, and LM(i) = 1 if $x_i \in TSu$. Clearly, LM is a binary sequence of length k. Denote k' = log2 $\lfloor(k + 1)\rfloor$, where $\lfloor\ \rfloor$ is the ceiling function. Take l = |i : LM(i) = 1| which is the amount of underflow/overflow blocks. Then we define a binary sequenceLM_c of length l_c = (l + 1)k' to record all underflow overflow locations.

1. (LMc(1),...,LMc(k)) is the binary representation of l.
2. For eachj ∈ {1,...,l},(LMc(jk'+1),...,LMc(jk'+k')) is the binary representation ofi, where i is the j -th index such that LM(i) = 1.

Step 3. Divide the k blocks in to three parts to get I_1,I_2 and I_3

1. $I_1 = \{X_1,...,X_{k1}\}$ with $k_1 = \lfloor lc/n \rfloor$
2. $I_2 = \{X_{k1+1},...,X_{k1+k2}\}$ such that it contains exactly l_c embeddable blocks. Here, a block is called embeddable if its value belongs to S_e.
3. $I_3 = \{X_{k1+k2+1},...,X_k\}$ is the set of rest blocks.

Step 4. Embed the hidden data into I_1 and I_3, i.e., for any i ∈ 1,...,k_1,k_1 + k_2 + 1,...,k

1. If $x_i \in T_s$, replace the value of X_i by $g(x_i)$.
2. If $x_i \in S_e$, replace the value of X_i by$f_m(x_i)$, where m ∈ {0,1} is the data bit to be embedded.
3. If $x_i \in TS_u$, we do nothing with X_i .

Step 5. Record LSBs of the first l_c pixels of I_1 to get a binary sequence S_{LSB} (recall that I_1 contains $nk_1 = n\lfloor lc/n \rfloor \geq l_c$ pixels). Then replace these LSBs by the sequence LM_c defined in Step 2.

Step 6. Embed the sequence S_{LSB} into I_2 in the same way as step 4. Since the length of S_{LSB} isl_c, S_{LSB} can be embedded exactly into the embeddable blocks ofI_2. Finally, we get the marked image.

D. *Extraction Process*

The data extraction procedure also contains several steps. Firstly similarto the data embedding process, divide the marked image blocks into three parts to getI_1, I_2 and I_3. Then, determine the location map by reading LSBs of I_1. Next, according to the location map and by using shifting and embedding functions, determinethe LSB sequence by extracting data from I_2, and then replace the LSBsof I_1 by the extracted LSB sequence. Finally, extract the embedded data from $I_1$and I_3. Notice that, using shifting and embedding functions, the image restorationcan be realized simultaneously with the data extraction. Extraction can be doneas a reversible process such that, from the technique, we can extract the hiddendata and the image which is needed as a cover image, separately. Integer WaveletTransform is done as a reversible process such that the transformed image can be retrievedback. The detailed data extraction procedure is given in algorithm 2.

Algorithm 2 Algorithm to perform data extraction

Step 1. The same as Step 1 of data embedding, divide the marked image into k non overlapping blocks $\{Y_1,...,Y_k\}$. Assume the value of Y_i is $y_i \in A_n$.

Step 2. Firstly, determine the amount of problematiclocations, by reading LSBs of the first k' = $\lfloor \log_2(k+1) \rfloor$ pixels. Secondly, read LSBs of the first l_c = (l + 1)k' pixels to get the sequence LM_c. Then we can get the location map LM. Finally, with $k_1 = \lfloor lc/n \rfloor$ k', LM, and by identifying embeddable blocks, we can obtain the same partition as defined in Step 3 of data embedding.

Step 3. Extract data from I_2 and recover original pixel values of I_2, i.e., for any i ∈ {k_1 + 1,...k_1 + k_2}.

a. If LM(i) = 0 and $y_i \in g(Ts)$, the original pixel value is $g-1(y_i)$ and there is no embedded data.
b. If LM(i) = 0 and $y_i \in f_m(S_e)$ holds for a certain m ∈ {0,1}, the original pixel value is $(f_m)-1(y_i)$ and the embedded data bit is m.
c. If LM(i) = 1, the original pixel value is $y_i$itself and there is no embedded data.

The sequence S_{LSB} defined in Step 5 of data embedding is extracted in this step.

Step 4. Replace LSBs of the first lc pixels of I_1 by S_{LSB}.

Step 5. Extract the embedded hidden data and recover original pixel values in $I_1 \cup I_3$ in the same way as Step 3.Finally, the hidden data is extracted and the original image is restored.

E. *Reversible Data Hiding (RDH)*

This subsection deals with how to embed auxiliary information and how to embed a required amount of data bits. As is known, a RDH algorithm usually depends on some parameters and these parameters should be communicated to decoder. To this end, slight modification to the embedding and extraction procedures is needed. Firstly, divide host image into two parts to get I_0

and I’.I_0 contains a fixed number of pixels and I’ is the rest pixels. Secondly, express the parameters in binary form to get a binary sequence and replace the LSBs of I_0 by this sequence. Here, the original LSBs of I_0 should be recorded and then embedded into host image as a part of hidden data. Finally, consider I’ as an image and embed data into it. For decoder, it only needs to read LSBs of I_0 to get the parameters, and then extract the hidden data fromI_0. In particular, I_0 can be restored by over writing its LSBs by a certain part of extracted hidden data. There are also a number of works on data hiding in the encrypted domain. The reversible data hiding in encrypted image is investigated in. Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain. This method by reserving room before encryption with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image. This method can achieve real reversibility, that is, data extraction and image recovery are free of any error. Thus the data hider can benefit from the extra space emptied out in previous stage to make data hiding process effortless. The proposed method can take advantage of all traditional RDH techniques for plain images and achieve excellent performance without loss of perfect secrecy.

F. Performance Analysis

Performance analysis is an assessment of process, equipment, employee, or anyother factor to gauge progress towards predetermined goals. It is guided and assessed by effectiveness, efficiency, and equity. A thorough performance analysis

- Identifies elements of effective management for performance under review.
- Develops required training and non-training performance improvement programs.
- Provides cost-effective ways for significant potential to improve performance.

The best use of performance analysis results in a consistent increase in performancerampup and productivity. It decreases the costs for developing and supporting desired performance. Here Mainly Three measures are used for performance analysis.

PSNR and MSE

PSNR is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR isusually expressed in terms of the logarithmic decibel scale. PSNR is most commonly used to measure the quality of reconstruction of lossy compression codecs (e.g., for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. When comparing compression codecs, PSNR is an approximation to human perception of reconstruction quality. Although a higher PSNR generally indicates that the reconstruction is of higher quality, in some cases it may not. One has to be extremely careful with the range of validity of this metric; it is only conclusively valid when it is used to compare results from the same codec (or codec type) and same content.PSNR is most easily defined via the mean squared error (MSE). Given a noise-free mXn monochrome image I and its noisy approximation K, MSE is defined as

$$MSE = {}^{1}\!/_{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - k(i,j)]^2 \qquad (1)$$

PSNR is defined as

$$\text{PSNR}=10 \log_{10}(MAX_i^2 \ / \ MSE) \qquad (2)$$

Here $MAX_i$is the maximum value of pixel in an image. In statistics, the MSE of an estimator measures the average of the squares of the”errors”, that is, the difference between the estimator and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn’t account for information that could produce a more accurate estimate.The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance of the estimator. Like the variance, MSE has the same units of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields the root- mean-square error or root-mean-square deviation (RMSE or RMSD), which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard deviation. And SSIM is defined as

$$\text{SSIM}=\frac{2\mu_x\mu_y+c1)(2\sigma_{xy}+c2))}{\mu_x^2+\mu_y^2+c1)(\sigma_x^2+\sigma_y^2+c2} \qquad (3)$$

μ_x and μ_y are the averages of x and y
σ_{xy} is the covariance of xy
$\sigma^2{}_x$ and $\sigma^2{}_Y$are the variances of x and y

III. MAIN IDEA OF HISTOGRAM-SHIFTING-BASED REVERSIBLE DATA HIDING [6]

In this method, the host image is divided into non overlapping blocks such that each block contains n pixels. Then, an n-dimensional histogram is generated by counting the frequency of the pixel-value-array sized no feach divided block. Finally, data embedding is implemented by modifying the resulting n-dimensional histogram. Notice that the pixel-value-array is an element of z^n we then needto divide z^n into two disjointed sets, one set is used tocarry hidden data by expansion embedding, and the other set is simply shifted to create vacant spaces to ensure the reversibility. We now present our new approach.

Let Sand T be a partition of z^n: S∪T =z^n and $S \cap T = \emptyset$.Suppose that three functionsg:T $\rightarrow z^n$ f_0:S$\rightarrow z^n$andf$_1$:$S \rightarrow z^n$satisfy the following conditions:
C1*:* The functions g, f0 and f1 are injective.
C2*:* The sets g(T), f_0(S)and f_1(S)are disjointed with each other.

Here, gis called “shifting function” and will be used to shift pixel values, f_0 and f_1 are called “embedding functions” and will be used to embed data. More specifically, each block with value **x** ϵTwill be shifted to g(**x**), and the block with value **x** ϵSwill be expanded to either f_0(**x**)orf$_1$(**x**)to carry one data bit. The shifting and embedding functions will give a HS-based RDH algorithm where the reversibility can be guaranteed by the conditions $C_1$and C_2.The underflow/overflow is an inevitable problem of RDH, i.e., for gray-scale image, the shifted and expanded values should be restricted in the range of [0, 255]. To deal with this, the above defined sets T and Sneed be further processed. Let

$$A_n = \{\mathbf{x} = (x_1, \ldots, x_n) \in z^n : 0 \le x_i \le 255\} \quad (4)$$

be the set of all pixel-value-arrays of length *n* of gray-scale image. We define

$$T_s = A_n \cap g^{-1}(A_n) \quad (5)$$

$$S_e = A_n \cap f_0^{-1}(A_n) \cap f_1^{-1}(A_n) \quad (6)$$

$$T_{u,o} = A_n \cap T - T_s (7)$$

$$S_{u,o} = A_n \cap S - S_e. \quad (8)$$

The sub-indices “s”,” e” and “u, o” mean “shift”, “embed “and “underflow/overflow”, respectively. Obviously, the four sets T_s, S_e, $T_{u,o}$ and $S_{u,o}$ are disjointed with each other and constitute a partition of A_n,

i.e.,$A_n = T_s \cup S_e \cup T_{u,o} \cup S_{u,o}$. (9)

Moreover, the sets g(Ts), $f_0(S_e)$ and $f_1(S_e)$are contained in A_n and the condition C_2 ensures that they are also disjointed. Each block with value **x** ϵTswill be shifted, each block with value **x** ϵS_e will be expanded to carry one data bit, and the block with value **x** ϵ $T_{u,o} \cup S_{u,o}$ will remain unchanged since it cannot be shifted or expanded due to underflow/overflow.

IV. Implementation Details

A medical image in JPEG format has to be selected for data embed- ding whose size has to be determined. If the image has a third plane, i.e. if it is a color image then it has to be converted to gray scale for the ease of processing. Then the locations that cause overflow and underflow when data is embedded in it need to be found out. These locations are specified based on the parameter “L”. Then P = 2L is identified after which the histogram of original image is generated. Shifted image is constructed by finding the locations in the image that is having pixel value < p and > (255 − p).After finding out these locations P is added to the former and P is subtracted from the latter. Histogram of shifted image is constructed next and the difference image is calculated. The direction of difference image calculation is shown in figure 4. Then the location map, which holds the underflow and overflow locations, is constructed. The entry of the location map is set to 1 to indicate that, that location consists of no hidden data. The location map also needs to be transmitted to the receiver. So the location map is also embedded in the host image. The encoded version of the location map is embedded in the image, otherwise there can be no space for the secret message embedding i.e. the secret message (confidential medical data) combined with location map data form the overall message that has to be hidden. The message is encrypted as well as compressed. Next, depending upon the algorithm developed shifting or embedding of the message is performed. The watermarked image thus obtained is transmitted to the receiver end. At the receiver end the reverse of operations performed at the sender side has to be done. Thus the image as well as the secret data can be recovered simultaneously and the performance evaluations using SSIM, PSNR, and MSE can be performed. The results presented in the following section shows the effectiveness of the method.

Main advantages are

- It prevent underflow and overflow
- Adopt a histogram shifting technique that narrows the histogram from both sides.
- Record the overhead bookkeeping information

In this case the location map contains the overhead information. And is defined as

$$\text{Location Map,} = \begin{cases} 0, if\ pixel\ \in [2L, 255\ -\ 2L] \\ \qquad 1, Otherwise \end{cases} \qquad (10)$$

Where L is an integer parameter used to narrow the histogram from both sides. When the value of L increases the embedding capacity increases. The embedding procedure is done by using the following Calculation
If di ≥ 2L shift x_i by 2L units

$$y_i = \begin{cases} x_i, \text{if } i = 0 \\ x_i + 2^L, \text{if } d_i \geq x_{i-1} \\ x_i - 2^L, \text{if } d_i \geq 2^L \text{ and } x_i < x - i - 1 \end{cases}$$

(11)

Where x_i and y_i are the original and watermarked value of pixel i.And d_i is the pixel difference. If di < 2L modify x_i according to the message bit, b

$$y_i = \begin{cases} x_i + (d_i + b), \text{if } x_i \geq x_{i-1} \\ x_i - (d_i + b), \text{ if } x_i < x_{i-1} \end{cases}$$

(12)

The extraction process is done by following Calculation

$$x_i = \begin{cases} y_i + \lceil \frac{|y_i - x - i - 1|}{2} \rceil, \text{If } |y_i - x_{i-1}| < 2^{L-1} and y_i < x_{i-1} \\ y_i - \lceil \frac{|y_i - x - i - 1|}{2} \rceil, \text{If } |y_i - x_{i-1}| < 2^{L-1} and y_i > x_{i-1} \\ y_i + 2^L, \text{ If } |y_i - x_{1-1}| \geq 2^{L-1} and y_i < x - i - 1 \\ \\ y_i - 2^L, \text{ If } |y_i - x_{1-1}| \geq 2^{L-1} and y_i > x - i - 1 \\ y_i, \text{Otherwise} \end{cases}$$

(13)

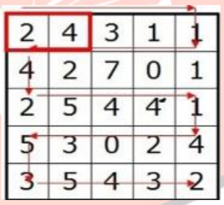
If $|y_i - x_{i-1}| < 2^{L+1}$, extract message bit, b by

$$b = \begin{cases} 0, \text{If } |y_i - x_i| \text{ is even} \\ 1, \text{If } |y_i - x_i| \text{ is odd} \end{cases}$$

(14)

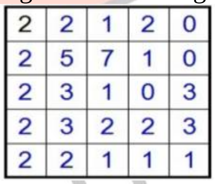
Where x_{i-1} denotes the restored value of y_{i-1}

V. EXAMPLE



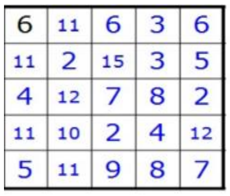
2	4	3	1	1
4	2	7	0	1
2	5	4	4	1
5	3	0	2	4
3	5	4	3	2

Figure 4-Host image



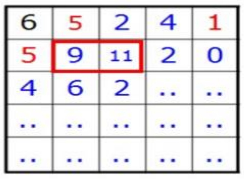
2	2	1	2	0
2	5	7	1	0
2	3	1	0	3
2	3	2	2	3
2	2	1	1	1

Figure 5-Pixel Difference



6	11	6	3	6
11	2	15	3	5
4	12	7	8	2
11	10	2	4	12
5	11	9	8	7

Figure 6- Watermarked Image



6	5	2	4	1
5	9	11	2	0
4	6	2	..	..
..	..	..	..	..
..	..	..	..	..

Figure 7- Calculation of y_i-$x_{(i-1)}$

2	4	3	1	1
4	2	7	0	1
2	5	4	4	1
5	3	0	2	4
3	5	4	3	2

Figure 8-Restored Image

VI. Performance Measurement

Table 1-PSNR values for selected images

Image	Embedding				Extraction			
	L=0	L=1	L=2	L=3	L=0	L=1	L=2	L=3
1	52.04	47.09	42.54	38.39	91.15	71.90	58.59	44.65
2	55.94	49.92	43.90	37.87	90.15	70.23	50.25	42.22
3	52.40	47.28	42.88	39.39	91.15	71.46	53.56	44.63
4	51.56	46.16	41.34	37.53	91.15	81.24	65.44	53.14
Average	52.99	47.61	42.67	38.30	90.90	73.71	56.96	46.16

Table 2- MSE values for selected images

Image	Embedding				Extraction			
	L=0	L=1	L=2	L=3	L=0	L=1	L=2	L=3
1	0.4057	1.2680	3.621	9.40	0.0020	0.0040	0.0898	2.2000
2	0.1655	0.6621	2.648	10.59	0.0165	0.0042	0.0600	1.9230
3	0.3734	1.2100	3.347	7.47	0.0020	0.0046	0.2861	2.2370
4	0.4535	1.5720	4.766	11.48	0.0020	0.0004	0.0180	0.3151
Average	0.3495	1.1780	3.5955	9.7340	0.0056	0.0033	0.1135	1.6688

Table 3-SSIM values for selected images

Image	Embedding				Extraction			
	L=0	L=1	L=2	L=3	L=0	L=1	L=2	L=3
1	0.9495	0.9160	0.8866	0.8485	0.9230	0.9990	0.9991	0.9931
2	0.8894	0.8776	0.8391	0.7400	0.8890	0.8290	0.8144	0.7932
3	0.9486	0.9244	0.8926	0.8668	0.8700	0.9764	0.9706	0.9555
4	0.9849	0.9614	0.9200	0.8688	0.9900	0.9994	0.9763	0.8481
Average	0.9431	0.9199	0.8846	0.8310	0.9180	0.9510	0.9401	0.8975

VII. Conclusion

In this paper, by revisiting existing algorithms, a general framework to construct HS-based RDH is proposed. According tothis framework, to obtain a RDH algorithm, one just needs to define the shifting and embedding functions. This work will facilitate the design of RDH. Furthermore, by incorporating our framework with the PEE and pixel selection techniques, one novel RDH algorithm is also introduced. This algorithm can achieve a better performance compared with the state-of-the-art works. So the proposed framework has a potential to provide excellent RDH algorithms.

References

[1] G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux, Reversible watermarking for knowledge digest embedding and reliability controlin medical images, IEEE Trans. Inf. Technol. Biomed., vol. 13, no. 2, pp. 158–165, Mar. 2009.

[2] H.-T. Wu and J. Huang, Reversible image watermarking on prediction errors by efficient histogram modification, Signal Process., vol. 92,no. 12, pp. 3000–3009, Dec. 2012.

[3] J. Tian, Reversible data embedding using a difference expansion, IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896,Aug.2003

[4] J. Zhou and O. C. Au, "Determining the capacity parameters in PEE based reversible image watermarking, IEEE Signal Process. Lett.,vol. 19, no. 5, pp. 287–290, May 2012.

[5] R. Li, O. C. Au, C. K. M. Yuk, S. Yip, and T. Chan, Enhanced image trans-coding using reversible data hiding, in Proc. IEEE Int. Symp.Circuits Syst., May 2007, pp. 1273–1276.
[6] Xiaolong Li, Bin Li, Bin Yang, and TieyongZeng General framework to histogram-shifting-based reversible data hiding.IEEE transaction on image processing vol 22 no 6 june 2013
[7] Y. Hu, H. K. Lee, and J. Li, DE-based reversible data hiding with improved overflow location map, IEEE Trans. Circuits Syst. VideoTechnol., vol. 19, no. 2, pp. 250–260, Feb. 2009

Appendix

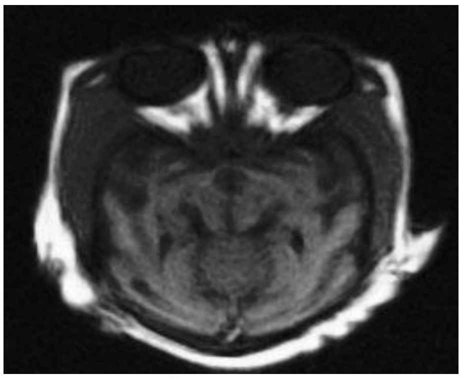


Figure 9- Image labeled as 1 in the table1,2,3

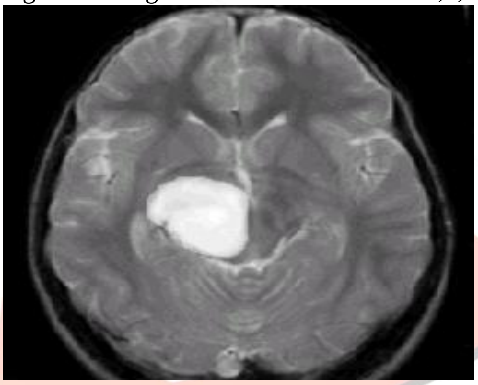


Figure 10- Image labeled as 2 in the table1,2,3

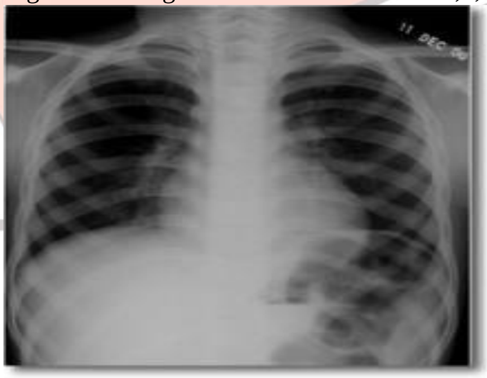


Figure 11- Image labeled as 3 in the table1,2,3

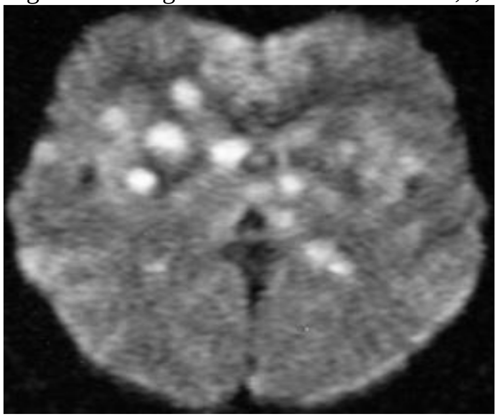


Figure 12- Image labeled as 4 in the table1,2,3