ISSN 2347 - 3983
Volume 8. No. 7, July 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter99872020.pdf
https://doi.org/10.30534/ijeter/2020/99872020

Improved Reversible Data Hiding in Medical images using Interpolation and Threshold based Embedding Strategy

M. Mahasree[1], N. Puviarasan[2], P. Aruna[3]
[1]Research Scholar, Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamilnadu, India, mahasree05@gmail.com
[2]Professor, Department of Computer and Information Science, Annamalai University, Chidambaram, Tamilnadu, India, npuvi2410@yahoo.in
[3]Professor, Department of Computer Science and Engineering, Annamalai University, Chidambaram, Tamilnadu, India, arunapuvi@yahoo.co.in

ABSTRACT

Reversible Data Hiding (RDH) techniques have become an active research topic in the field of Data Security. The two important factors in defining the efficiency of data hiding method are stego quality and payload capacity. Recently interpolation based data hiding techniques (IRDH) are useful for achieving high embedding capacity by up-scaling the carrier image to create additional space for embedding. Existing interpolation-based hiding techniques concentrate mainly on natural images. But IRDH can also be utilized for sensitive medical images to hide patient details inside them for achieving privacy protection. Thus, this paper encompasses an effective embedding procedure for interpolation based reversible data hiding schemes in medical images. The objective of our proposed scheme is to provide high visual quality as well as increased payload. The given medical image is interpolated without changing its original pixels. For embedding patient details, we propose a novel Threshold-based Embedding Strategy (TES) to increase the number of bits that can be embedded in an interpolated pixel. Experimental results show that the proposed interpolation based reversible data hiding technique achieves high stego quality and capacity than many state-of-the-art methods for medical images.

Key words: Reversible Data Hiding (RDH), Interpolation based Reversible Data Hiding (IRDH), Threshold-based Embedding Strategy (TES)

1. INTRODUCTION

Data Hiding has been used to securely share sensitive and private information over internet [1], [2]. Data Hiding is categorized into two types. One is Reversible data hiding (RDH) and the other is Irreversible data hiding (IDH) [3], [4]. Traditionally, RDH works in four domains namely compression [5]-[7], histogram modification, quantization and expansion [8]. Recently, Interpolation-based Reversible Data Hiding (IRDH) Schemes are being developed for high embedding capacity. Interpolation is one of the image processing techniques, used to enlarge an image by creating new pixels. It exploits the dependency of neighborhood pixels to create a new pixel. The most common interpolation techniques used in imaging are Nearest Neighbor Interpolation (NNI) and Bilinear Interpolation (BI). But the scaled up images obtained from these traditional methods does not satisfy the image quality needed for data hiding schemes [9]. So, various interpolation schemes like Neighbor Mean Interpolation (NMI), Interpolation by Neighboring Pixels (INP), Enhanced Neighbor Mean Interpolation (ENMI) etc., have been developed for reversible data hiding [9], [10]. However, these methods suffer from capacity limitations. The interpolated pixels are not effectively utilized for data embedding which leads to low embedding capacity. Therefore, there is possibility for increasing the payload.

In this paper, an improved interpolation based reversible data hiding is proposed for hiding patient records into medical images. Interpolation by Neighboring Pixels is used to interpolate the given image. This up-scaled image is the cover image. Based on a threshold value, the number of bits to be embedded in each interpolated pixel is calculated using the proposed Threshold-based Embedding Strategy (TES) technique.

The remainder of this paper is organized as follows. Section 2 discusses the related works in IRDH. Section 3 presents the proposed system in detail. Section 4 shows the experimental results. Conclusions are provided in Section 5.

2. RELATED WORKS

Jung and Yoo first introduced the use of interpolation in reversible data hiding. They proposed an interpolation method namely Neighbor Mean Interpolation (NMI) to embed secret data into those interpolated pixels [11]. Lee and Huang presented a better interpolation method to NMI which is named as Interpolation by Neighboring Pixels (INP). This method provides better visual quality then Jung and Yoo's method [12]. Tang et al. proposed a better RDH namely CRS, using interpolation to increase the hiding capacity [13]. Hu and Li extended the INP method using Interpolation by

Maximizing the difference values between Neighboring Pixels (IMNP). This technique improved the embedding capacity of Lee and Huang's method [14]. Lu proposed an interpolation based lossless hiding using message recoding step. Here, INP interpolation is used. The re-encoding scheme assigns smaller codes to the secret message based on its frequency of occurrence [15].

Mohammad et al. presented a new and computationally simple reversible scheme based on interpolation. In this work, the number of embeddable bits and stego quality is maintained by threshold value [16]. Luo et al. proposed a reversible watermarking scheme based on interpolation-error expansion. The additive interpolation is followed by difference expansion of the interpolated pixels [17]. Wang et al. exploited the interpolation technique and direction order mechanism. Interpolation errors are calculated for one type of pixels called wall pixels whereas difference values are calculated for other type of pixels called non-wall pixels. Embedding is carried out based on direction order and histogram shifting [18]. Biswapati et al. proposed a reversible data hiding scheme based in a weighted matrix. Modulus function and element-wise multiplication steps are employed to hide secret data [19]. Govind and Wilscy presented a new RDH with directional interpolation and difference expansion using a threshold value [20].

3. PROPOSED SYSTEM

In this section we present our novel reversible data hiding using interpolation and Threshold-based Embedding Strategy (TES). The general structure of interpolation based reversible data hiding scheme is shown in Figure 1. The proposed RDH proceeds by the application of INP method. Then, the interpolated pixels are processed in 3 x 3 blocks. Based on the difference value between interpolated pixels and neighbor pixels, payload capacity of each interpolated pixel is calculated using TES strategy. Then the bits are embedded into the interpolated pixels without degrading the overall visual quality of the stego image. Next, the extraction algorithm is explained where both cover and secret data are separated from the stego image reversibly.

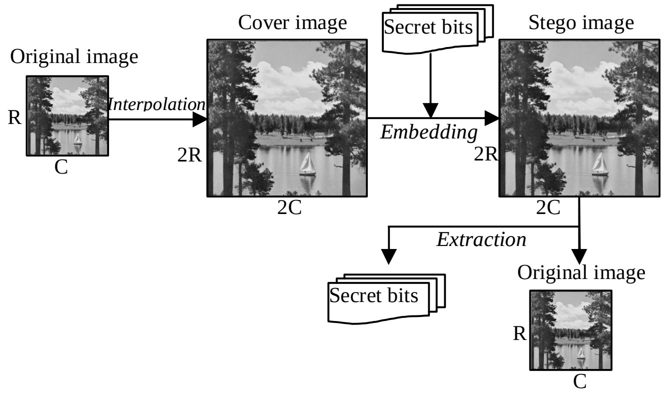


Figure 1: Framework for Interpolation-based Reversible Data Hiding Techniques

3.1 Interpolation

Image interpolation is a technique by which low resolution image can be scaled up to high resolution image without degrading the visual quality. The quality of such interpolated images can be calculated using PSNR value. But the enlarged images are larger in size than the given original images. To solve this problem, we consider the standard ground truth images of size 512 × 512 and reduce them to 256 × 256. This reduced image serves as the original image and is further up-scaled using interpolation technique to generate cover image of size equal to the ground truth image. Now, we can calculate the performance of interpolation method by finding the distortion between ground truth image and cover image using PSNR metric.

Table 1: PSNR comparisons of NMI, INP, CRS and IMNP methods for interpolated image and ground truth image

Images	NMI	INP	CRS	IMNP
Airplane	34.93	35.29	34.15	34.09
Baboon	29.92	30.08	29.72	29.55
Barbara	31.87	32.10	31.49	31.34
Goldhill	32.78	33.15	32.25	31.96
Lena	34.63	35.13	33.94	33.53
Peppers	33.69	34.01	33.20	32.94

Table 1 compares the performance of the NMI, INP, CRS and IMNP methods to determine an interpolation method that is appropriate for our proposed method. From the table, it is evident that INP interpolation gives better result than other interpolation methods. Hence, INP is preferred in our proposed data hiding technique.

Let $R(i,j)$ be the original image of size 256 × 256 and $C(i,j)$ be the cover image of size 512 × 512 obtained from original image using INP interpolation. The scaling procedure is described as follows. As pointed out in Figure 2, for each 2 × 2 block of original image, five new pixels are inserted to form a 3 × 3 block. The corner pixels of cover image *C(0,0)*, *C(0,2)*, *C(2,0)* and *C(2,2)* are the unaltered original image pixels *R(0,0)*, *R(0,1)*, *R(1,0)* and *R(1,1)* respectively as shown in the Figure 2. The middle pixels are calculated as follows:

$$C(0,1) = \left\lfloor \frac{C(0,0) + \frac{C(0,0)+C(0,2)}{2}}{2} \right\rfloor \; ; \; C(1,2) = \left\lfloor \frac{C(0,2) + \frac{C(0,2)+C(2,2)}{2}}{2} \right\rfloor$$

$$C(1,0) = \left\lfloor \frac{C(0,0) + \frac{C(0,0)+C(2,0)}{2}}{2} \right\rfloor \; ; \; C(2,1) = \left\lfloor \frac{C(2,0) + \frac{C(2,0)+C(2,2)}{2}}{2} \right\rfloor$$

$$C(1,1) = \left\lfloor \frac{C(0,1) + C(1,0)}{2} \right\rfloor \quad (1)$$

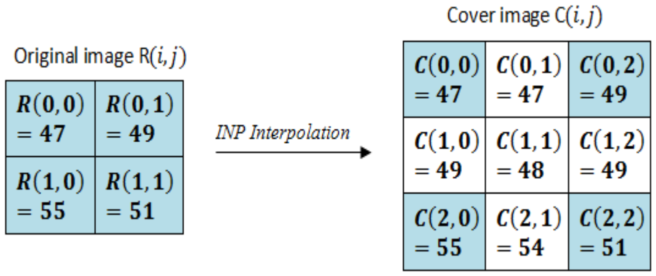


Figure 2: Generation of cover image from original image using INP

3.2 Embedding Algorithm using TES

After interpolation, the next step is embedding. The proposed Threshold-based Embedding Strategy (TES) exploits the relationship among predicted pixels and original pixels of the cover image. The steps for embedding are given below in detail.

Input: Cover Image *C* sized *2r* × *2c* and secret bit stream *s*
Output: Stego Image *S* sized *2r* × *2c*

Step 1: Divide Cover into *2* × *2* non-overlapping blocks. Calculate the maximum of *2* × *2* block denoted as *M.*

$$M = \max(C(i,j), C(i,j+1), C(i+1,j), C(i+1,j+1)) \qquad (2)$$

Step 2: Find the difference values $d_k (k = 1,2,3)$ between the maximum value *M* and the interpolated pixels *C(i,j+1), C(i+1,j)* and *C(i+1,j+1)* as given in eq. (3). At each block, the first pixel *C(i,j)* is kept unchanged for distortion-less recovery. The remaining pixels *C(i,j+1), C(i+1,j)* and *C(i+1,j+1)* are used for embedding.

$$\begin{aligned} d_1 &= \mathrm{M} - C(i,j+1) \\ d_2 &= \mathrm{M} - C(i+1,j) \\ d_3 &= \mathrm{M} - C(i+1,j+1) \end{aligned} \qquad (3)$$

Step 3: Obtain the lower and upper bounds to determine the total bits to be embedded. The log base 2 values of d_k are denoted as l_k . The equations are as follows.

$$\begin{aligned} l_1 &= \lfloor log_2(\mathrm{d}_1) \rfloor \\ l_2 &= \lfloor log_2(\mathrm{d}_2) \rfloor \\ l_3 &= \lfloor log_2(\mathrm{d}_3) \rfloor \end{aligned} \qquad (4)$$

Step 4: Allocate the maximum number of embeddable bits to be inserted into an interpolated pixel using threshold value *th*. This threshold value is the user defined value to choose between stego quality and capacity.

$$n_k = \begin{cases} th, & if\ l_k = \infty \\ th+1, & if\ l_k \in \{0,1,2,3\} \\ th+2, & if\ l_k \in \{4,5\} \\ th+3, & if\ l_k \in \{6,7\} \end{cases} \qquad (5)$$

where k=1,2,3 denotes the interpolated pixels *C(i,j+1), C(i+1,j)* and *C(i+1,j+1)* respectively. The value of n_k denotes the length of secret bits.

Step 5: Convert n_k into decimal number and add with the cover pixels to form the stego pixels. But this may lead to overflow condition where the stego pixel value exceeds the maximum intensity of 255. So, to overcome this situation, limitation variable *R* is introduced.

$$R = 255 - (2^{(th+3)} - 1) \qquad (6)$$

Let s_k be the bit string of length n_k. Therefore, the embedding equations are modified as follows.

$$S(i,j) = C(i,j)$$

$$S(i,j+1) = \begin{cases} C(i,j+1) + decimal(s_1), & if\ C(i,j+1) \le R \\ C(i,j+1) - decimal(s_1), & otherwise \end{cases}$$

$$S(i+1,j) = \begin{cases} C(i+1,j) + decimal(s_2), & if\ C(i+1,j) \le R \\ C(i+1,j) - decimal(s_2), & otherwise \end{cases}$$

$$S(i+1,j+1) = \begin{cases} C(i+1,j+1) + decimal(s_3), & if\ C(i+1,j+1) \le R \\ C(i+1,j+1) - decimal(s_3), & otherwise \end{cases} \qquad (7)$$

Step 6: Go back to step 2 and repeat the same process until all the non-overlapping blocks are processed. Finally, we get the Stego-image *S*

3.3 Illustration of Embedding Algorithm

Consider the interpolated 2 × 2 block as shown in Figure 2. The cover pixels are *C(0,0)=47, C(0,1)=47, C(1,0)=49, C(1,1)=48*. Find the maximum of the four pixels, *M=49*. Next step is to find the difference values, d_1 = 49-47 = 2, d_2 = 49-49 = 0, d_3 = 49-48 = 1. The difference values are in varied range from 0 to 255. So, to group the interpolated pixels and allocate the capacity, log values are calculated. $l_1 = \lfloor log_2(2) \rfloor = 1$, $l_2 = \lfloor log_2(0) \rfloor = \infty$, $l_3 = \lfloor log_2(1) \rfloor = 0$. Based on our proposed bit allocation method, *th* values are selected, $n_1 = th + 1$, $n_2 = th$ and $n_3 = th + 1$. The *th* values are used defined. The next step is to check overflow or underflow limitations before embedding. The limitation variable *R* depends on the threshold value. Let us assume $th = 1$, then *R*=255-(2$^{(1+3)}$-1) = 255-(16-1) = 240. Let us consider secret message bits *s*=01110. So, $n_1 = 2$, $n_2 = 1$ and $n_3 = 2$. The value of $n_1 = 2$ indicates that first two bits from *s* has to be taken and converted to integer value, i.e., *decimal* (n_1) =*decimal (01)* =*1*. Similarly, *decimal*(n_2)=*decimal(1)=1* and *decimal*(n_3)=*decimal(10)=2*. According to eq. 6 all three interpolated pixels are within range and hence stego pixels are formed by adding integer values to interpolated pixels, *S(0,0)=47, S(0,1)=47+1=48, S(1,0)=49+1=50* and *S(1,1)=48+2=50*. This example is explained in Figure 3.

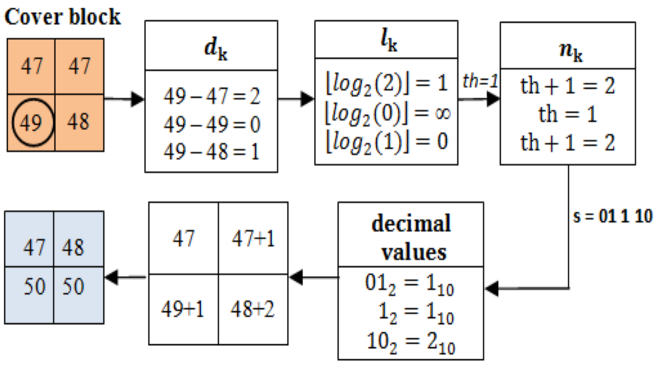


Figure 3: Illustration of Proposed TES method with example

3.4 Extraction Algorithm

At the receiver side, to extract the secret data from the stego image and to recover the original image, the proposed extraction algorithm is applied. The detail of the extraction algorithm is given below.

Input: Stego Image *S* sized *2r* × *2c*

Output: Original Image R sized *r* × *c* and secret bit stream *s*

Step 1: Divide Stego into 2 × 2 non-overlapping blocks. Collect all the first pixels *S(i,j)* from each block to get the Original Image *R*.

$$O(i,j) = S(2i,2j)\ for\ all\ i,j\ \in \{0,255\} \qquad (8)$$

Step 2: Use the INP interpolation to get the interpolated image from the recovered Original image (Eq.1)

Step 3: Find the absolute difference between the interpolated image and the Stego image. The difference value is the value of integer hidden in stego image**.**

$$Diff[n] = abs(S(x,y) - C(x,y)) \qquad (9)$$

Step 4: Follow the steps of embedding (Eq. 2, 3, 4, 5) to find the n_k value for each block. Convert the *Diff[n]* integer into binary bits of length n_k.

Step 5: Extract all the secret bits and concatenate them to regain the secret data losslessly.

3.5 Illustration of Extraction Algorithm

At the extraction side, the original image is separated from the cover image by colleting the unchanged pixel *S(i,j)* in every 2 × 2 block. In our example, the first pixel 47 is the original pixel. Now, same INP Interpolation is performed on recovered original image. By finding the difference matrix between the interpolated image and the stego image, secret bit stream is recovered losslessly as shown in Figure 4.

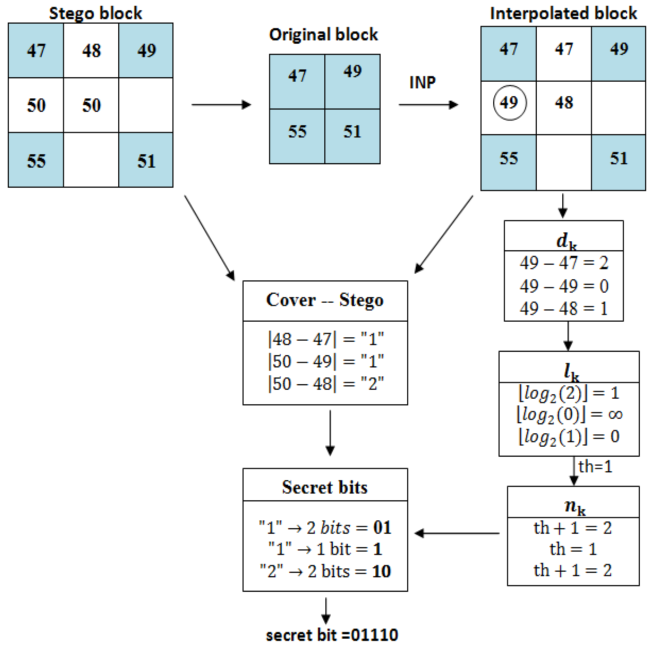
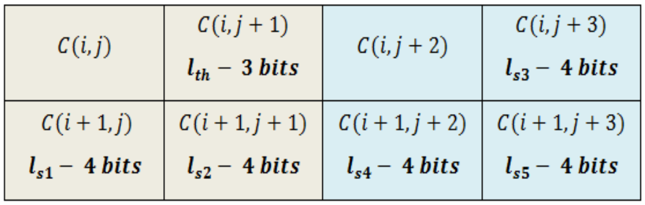


Figure 4: Illustration of Proposed Extraction method with example

3.6 Auxiliary information

The auxiliary information needed for extraction are the length of total secret bits embedded and the threshold value. To represent the secret data length, we need a maximum of 20 bits. The maximum value of possible threshold value is 5 and hence it requires 3 bits. So, the additional data of l_{th}+l_s is embedded into the first two blocks of cover image using LSB substitution as shown in Figure 5. Then, the proposed embedding starts with the third block and proceeds till all required bits are hidden successfully.



$C(i,j)$	$C(i,j+1)$ l_{th} − **3 bits**	$C(i,j+2)$	$C(i,j+3)$ l_{s3} − **4 bits**
$C(i+1,j)$ l_{s1} − **4 bits**	$C(i+1,j+1)$ l_{s2} − **4 bits**	$C(i+1,j+2)$ l_{s4} − **4 bits**	$C(i+1,j+3)$ l_{s5} − **4 bits**

Figure 5: Embedding of Auxiliary information using LSB substitution in first two 2 × 2 blocks

4. EXPERIMENTAL RESULTS

This section presents the experimental results and comparison of the proposed method with other state-of-art methods. Experiments were done on 8-bit grayscale medical images of size 512 x 512. These images serve as ground truth. The interpolated cover image is compared with ground truth (input image) image to show the performance of the interpolation scheme. So, the original image (carrier of secret data) is formed by reducing the input image to ¼ of its size. It is to be noted that for all the existing methods taken for comparison and the proposed method we used Bicubic Interpolation for this reduction step. The image quality of the embedded image can be evaluated using PSNR as follows:

$$PSNR = 10 * log\left(\frac{255^2}{MSE}\right) \qquad (10)$$

$$MSE = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}\big(S(i,j) - I(i,j)\big)^2 \qquad (11)$$

where PSNR is Peak Signal-to-Noise Ratio and MSE is Mean Square Error between Stego image *S(i,j)* and Input image *I(i,j)* with dimension *M* x *N*. Structural Similarity Index Measure is given by,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

$$c_1 = (k_1L)^2, \quad c_2 = (k_2L)^2 \qquad (12)$$

where μ_x and μ_y are average values; σ_x^2 and σ_y^2 denotes variance, σ_{xy} is the covariance of x and y; c_1 and c_2 are variables for stabilization, L=255 denotes pixel range, k_1=0.01, k_2=0.03 are default constants and x, y are stego and input images respectively. Embedding capacity denotes the total number of bits that can be embedded into the cover image. It is named as BPP (bits per pixel). The secret data are the randomly generated binary bits.

Table 2: Comparison of Proposed method with existing IRDH [12] for Image_3

Range of n_k	**Lee and Huang [12]**				**Proposed TES *(th=1)***			
	Number of pixels with n_k	**Number of bits to be embedded per pixel**	**Total bits embedded for a given n_k**	**Total Capacity (in bits)**	**Number of pixels with n_k**	**Number of bits to be embedded per pixel**	**Total bits embedded for a given n_k**	**Total Capacity (in bits)**
∞	120495	0	0	0	151961	*th*	151961	151961
0	12684	0	0		11484	*th+1*	22968	79466
1	17058	1	17058	155139	11729	*th+1*	23458	
2	18711	2	37422		9980	*th+1*	19960	
3	14671	3	44013		6540	*th+1*	13080	
4	8945	4	35780		3455	*th+2*	10365	14313
5	3400	5	17000		1316	*th+2*	3948	
6	642	6	3852		143	*th+3*	572	572
7	2	7	14		0	*th+3*	0	
				155139				246312

Table 3: Performance of the proposed method with different thresholds

Cover Images	**Proposed *th*=1**			**Proposed *th*=2**			**Proposed *th*=3**		
	PSNR	**BPP**	**SSIM**	**PSNR**	**BPP**	**SSIM**	**PSNR**	**BPP**	**SSIM**
Image_1	37.22	0.90	0.9179	36.11	1.65	0.7858	33.64	2.40	0.6455
Image_2	36.15	0.91	0.9147	35.33	1.66	0.8020	33.26	2.41	0.6855
Image_3	39.73	0.93	0.9481	37.80	1.68	0.8245	34.22	2.43	0.6878
Image_4	35.46	0.94	0.9008	34.79	1.69	0.8006	32.98	2.44	0.6911
Image_5	36.56	1.11	0.9391	35.14	1.86	0.8312	32.66	2.61	0.6333
Image_6	35.98	1.11	0.9622	34.85	1.86	0.9253	32.45	2.61	0.8167
Image_7	38.61	1.21	0.9421	36.40	1.96	0.8880	32.43	2.71	0.7519
Image_8	36.26	1.17	0.9545	34.94	1.92	0.9055	32.29	2.67	0.7736

Table 4: Comparison of the proposed method with existing techniques (th=1)

Cover Images	**Jung and Yoo [11]**		**Lee and Huang [12]**		**Mohammad et al. [16]**		**Proposed *th=1***	
	PSNR	**BPP**	**PSNR**	**BPP**	**PSNR**	**BPP**	**PSNR**	**BPP**
Image_1	37.08	0.38	36.67	0.64	36.13	0.51	37.22	0.90
Image_2	36.00	0.48	35.61	0.78	35.09	0.64	36.15	0.91
Image_3	39.49	0.29	38.64	0.59	37.84	0.41	39.73	0.93
Image_4	35.27	0.55	34.93	0.89	34.56	0.71	35.46	0.94
Image_5	36.44	0.55	35.76	0.99	35.15	0.74	36.56	1.11
Image_6	35.94	0.54	35.42	1.00	34.96	0.74	35.98	1.11
Image_7	38.37	0.47	37.46	1.12	36.63	0.71	38.61	1.21
Image_8	35.97	0.62	35.24	1.17	34.49	0.87	36.26	1.17

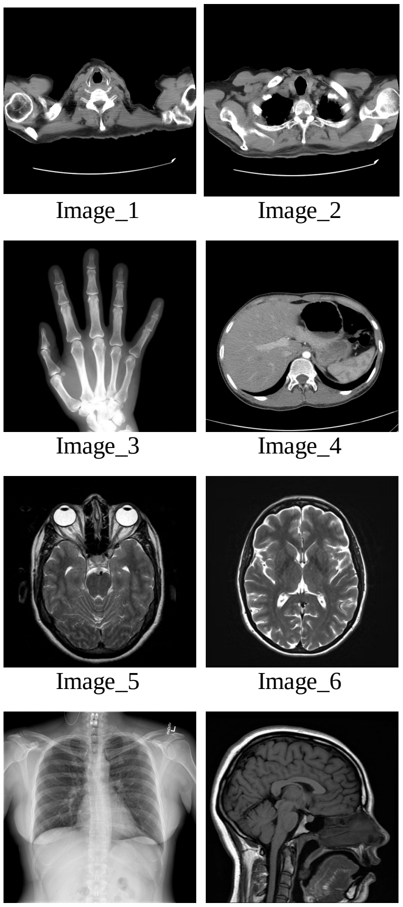


Image_1 Image_2

Image_3 Image_4

Image_5 Image_6

Image_7 Image_8

Figure 6: Groundtruth medical images

The test medical images which serve as groundtruth were downloaded from National Biomedical Imaging Archive database as shown in Figure 6. Table 2 shows how the proposed system is efficient when compared to existing methods [12]. Here, the number of bits to be embedded per pixel using threshold is our proposed way to increase the embedding capacity. It is to be noted that though both our proposed and existing system Lee and Huang [12] uses INP interpolation, the number of pixels with $n_k = \infty$ is higher for our proposed system. This is because we have proposed the equation for finding the difference values between interpolated and reference pixels using M value as in eq.2 and eq.3.

From the Table 2, it is evident that higher capacity is due to the increase in number of pixels with n_k values 0 and infinity. Minimum bits are selected for those pixels with higher occurrences of n_k to maintain the stego image visibility.

Table 3 shows the performance of different threshold values for the proposed TES method. The value of *th* is restricted between 1 and 3 to balance the tradeoff between capacity and PSNR. Thus the user can define the *th* value according to their need.

Table 4 shows the PSNR and capacity comparison of the proposed scheme with other existing techniques. The capacity of the proposed technique is higher than all other methods. Despite this increment, our PSNR values better than other existing techniques with lower capacity.

Table 5: Comparison of the proposed method with existing techniques *(th=2)*

Cover Images	**Tang et al. [13]**		**Proposed *th=2***	
	PSNR	**BPP**	**PSNR**	**BPP**
Image_1	35.56	0.79	36.11	1.65
Image_2	34.59	0.97	35.33	1.66
Image_3	37.12	0.72	37.80	1.68
Image_4	34.03	1.08	34.79	1.69
Image_5	34.33	1.23	35.14	1.86
Image_6	34.42	1.24	34.85	1.86
Image_7	35.27	1.40	36.40	1.96
Image_8	33.83	1.46	34.94	1.92

Table 5 shows the result of comparison between the existing [8] and proposed method with *th=2*. Tang method is better than other existing techniques so it is compared with higher threshold value to prove the novelty of the proposed method.

5. CONCLUSION

In this paper, a novel Threshold-based Bit Embedding Strategy (TES) is proved to be efficient for interpolation based information hiding. Interpolation by Neighboring Pixels is preferred for up-scaling the cover image. Then, the proposed embedding schemes allowed high payload to be embedded inside the cover image without disturbing the original pixels. Proposed method has an advantage of not using any location map or secret pin to be shared prior to communication. Also, the extraction of cover image is blind technique i.e. the cover image is extracted without any procedures or additional information. Experimental results showed the efficiency of the proposed technique suitable in medical applications where patient records can be securely saved into their medical images. PSNR, SSIM and BPP values also demonstrated the efficiency of the proposed TES based reversible information hiding techniques.

REFERENCES

1. T. Lu. **Adaptive (k, F1) interpolation-based hiding scheme**, *Multimedia Tools and Applications, Springer*, Vol. 76, Issue: 2, pp. 1827-1855, 2017. https://doi.org/10.1007/s11042-015-3168-2
2. B. Kusuma Priya, L. P. Maguluri, T. Srinivasarao, T. E. Rao. **A Systematic Approach for Data Hiding Using Cryptography and Steganography**, *International*

Journal of Emerging Trends in Engineering Research (IJETER), Vol. 8, Issue: 4, pp. 1326-1332, 2020. https://doi.org/10.30534/ijeter/2020/63842020

3. M. Mahasree, N. Puviarasan, P Aruna. **An Improved Reversible Data Hiding Using Pixel Value Ordering and Context Pixel-Based Block Selection**, in *Lecture Notes in Networks and Systems,* Singh P., Pawłowski W., Tanwar S., Kumar N., Rodrigues J., Obaidat M., Eds. Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019). Springer, Singapore, 2020, Vol. 121, pp 873-887. https://doi.org/10.1007/978-981-15-3369-3_64
4. M. Mahasree, N. Puviarasan, P Aruna. **Pixel Value Ordering Based Reversible Data Hiding with Novel MPBS Strategy**, *International Journal of Engineering and Advanced Technology (IJEAT)*, Vol. 9, Issue: 3, pp. 1518-1524, 2020.
5. K. Hema, C. Mylara Reddy, S. M. Rajesh, L. Kamala. **Efficient Image Compression by Machine Learning**, *International Journal of Emerging Trends in Engineering Research (IJETER)*, Vol. 8, Issue: 5, pp. 1672-1677, 2020. https://doi.org/10.30534/ijeter/2020/29852020
6. D. J. Ashpin Pabi, M. Mahasree, P. Aruna, N. Puviarasan. **Image Compression based on DCT and BPSO for MRI and Standard Images**, *Int. Journal of Engineering Research and Application (IJERA*), Vol. 6, Issue: 10, pp. 24-31, 2016.
7. M. Mahasree, D. J. Ashpin Pabi, P. Aruna, N. Puviarasan. **Adoption of Global Structure Transformation in lossy image compression based on Curvelet and Cosine transforms**, *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, Vol. 5, Issue: 3, pp. 4038-4051, 2017.
8. J. Shukla, M. Shandilya. **A Recent Survey on Information-Hiding Techniques**, *in Data, Engineering and Applications*, R. Shukla, J. Agrawal, S. Sharma, G. Singh Tomer, Eds. Springer, Singapore, 2019, pp. 57-70.
9. A. Malik, G. Sikka, H. K. Verma. **Image interpolation based high capacity reversible data hiding scheme**, *Multimedia Tools and Applications*, Springer, Vol.: 76, Issue: 22, pp. 24107-24123, 2017. https://doi.org/10.1007/s11042-016-4186-4
10. M. A. Wahed, H. Nyeem. **Reversible data hiding with interpolation and adaptive embedding**, *Multimedia Tools and Applications*, Springer, Vol. 78, Issue: 8, pp. 10795-10819, 2019.
11. K. Jung, K. Yoo. **Data hiding method using image interpolation**, *Computer Standards & Interfaces*, Elsevier, Vol. 31, Issue: 2, pp. 465-470, 2009.
12. C. Lee, Y. Huang. **An efficient image interpolation increasing payload in reversible data hiding**, *Expert Systems with Applications*, Elsevier, Vol. 39, Issue: 8, pp. 6712-6719, 2012. https://doi.org/10.1016/j.eswa.2011.12.019
13. M. Tang, J. Hu, W. Song. **A high capacity image steganography using multi-layer embedding**, *Optik*, Elsevier, Vol. 125, Issue: 15, pp. 3972-3976, 2014.
14. J. Hu, T. Li. **Reversible steganography using extended image interpolation technique**, *Computers & Electrical Engineering*, Elsevier, Vol. 46, pp. 447-455, 2015.
15. T. Lu. **An interpolation-based lossless hiding scheme based on message recoding mechanism**, *Optik*, Elsevier, Vol. 130, pp. 1377-1396, 2017.
16. A. A. Mohammad, A. Al-Haj, M. Farfoura. **An improved capacity data hiding technique based on image interpolation**, *Multimedia Tools and Applications*, Springer, Vol. 78, Issue: 6, pp. 7181-7205, 2019. https://doi.org/10.1007/s11042-018-6465-8
17. L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong. **Reversible Image Watermarking Using Interpolation Technique**, *IEEE Transactions on Information Forensics and Security*, Vol. 5, Issue: 1, pp. 187-193, 2010.
18. X. Wang, C. Chang, T. Nguyen, M. Li. **Reversible data hiding for high quality images exploiting interpolation and direction order mechanism**, *Digital Signal Processing*, Elsevier, Vol. 23, pp. 569-577, 2013. https://doi.org/10.1016/j.dsp.2012.06.015
19. J. Biswapati, G. Debasis, M. S. Kumar. **Weighted Matrix Based Reversible Data Hiding Scheme Using Image Interpolation**, *in International Conference on Computational Intelligence in Data Mining (CIDM)*, Vol. 2, pp. 239-248, 2015.
20. P. V. S. Govind, M. Wilsey. **A New Reversible Data Hiding Scheme with Improved Capacity Based on Directional Interpolation and Difference Expansion**, *in International Conference on Information and Communication Technologies (ICICT 2014)*, pp. 491-498, 2015. https://doi.org/10.1016/j.procs.2015.02.073