RESEARCH ARTICLE OPEN ACCESS

Image Compression based on DCT and BPSO for MRI and Standard Images

D.J. Ashpin Pabi[1], M. Mahasree[2], P. Aruna[3], N.Puviarasan[4]

[1]*(Research Scholar, Dept. of CSE, Annamalai University, Chidambaram, Tamil Nadu, India*
[2]*(PG Scholar, Dept. of CSE, Annamalai University, Chidambaram, Tamil Nadu, India*
[3]*(Professor, Dept. of CSE, Annamalai University, Chidambaram, Tamil Nadu, India*
[4]*(Associate Professor, Dept. of CSE, Annamalai University, Chidambaram, Tamil Nadu, India*

ABSTRACT

Nowadays, digital image compression has become a crucial factor of modern telecommunication systems. Image compression is the process of reducing total bits required to represent an image by reducing redundancies while preserving the image quality as much as possible. Various applications including internet, multimedia, satellite imaging, medical imaging uses image compression in order to store and transmit images in an efficient manner. Selection of compression technique is an application-specific process. In this paper, an improved compression technique based on Butterfly-Particle Swarm Optimization (BPSO) is proposed. BPSO is an intelligence-based iterative algorithm utilized for finding optimal solution from a set of possible values. The dominant factors of BPSO over other optimization techniques are higher convergence rate, searching ability and overall performance. The proposed technique divides the input image into 8×8 blocks. Discrete Cosine Transform (DCT) is applied to each block to obtain the coefficients. Then, the threshold values are obtained from BPSO. Based on this threshold, values of the coefficients are modified. Finally, quantization followed by the Huffman encoding is used to encode the image. Experimental results show the effectiveness of the proposed method over the existing method.

***Keywords**:* BPSO, DCT, Huffman encoding, quantization, threshold

I. INTRODUCTION

The main aim of image compression systems is to reduce the memory required to store the images; and transfer the images over longer distances with reduced cost and time. In some of the applications of security, reducing the transfer time may also decrease the chances of security attacks. The compression can be achieved when there are redundancies in the image. Redundancy is the term which denotes the presence of some irrelevant or repeated data in an image. Generally, there are three types of redundancy occurs in images [1]. They are 1) Coding redundancy: It emphasizes the process of bit allocation. When the codes (pixel values) are allotted with more bits than actually required, coding redundancy occurs. 2) Inter-pixel redundancy: Most of the images have neighboring pixels with highly correlated values, in other words, there are some larger regions in an image where the pixel values are almost the same. This is known as inter-pixel redundancy. 3) Psycho-visual redundancy: There are some slight intensity variations in images that cannot be differentiated by human eye. These visual variations which are less relevant to observer are called psycho-visual redundancy. To achieve compression these redundancies has to be removed.

Fig.1 shows the standard image compression system [2]. It is composed of two functional components: an encoder and a decoder. The original image as $f(i,j)$ is fed into the encoder of the compression system. The encoder removes the redundancies through a series of three independent operations 1) Mapper: It transforms the input image into a non-visual format designed to reduce inter-pixel redundancy 2) Quantizer: It quantizes the pixel values by eliminating less relevant information while preserving highly sensitive information 3) Symbol coder: It generates a fixed or variable length code to represent the quantized output and maps the output in accordance with the code. By doing so, it reduces coding redundancy. The decoder contains three components: symbol decoder, dequantizer and inverse mapper. They perform the reverse operations of encoder and reconstructed image $f'(i,j)$ is obtained as the output.

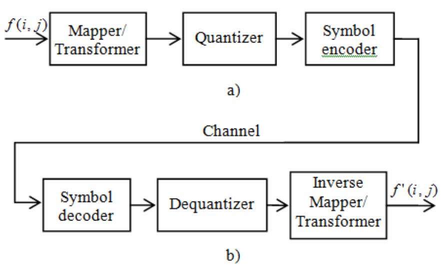


Fig.1. Block diagram of an image compression system: a) Encoder b) Decoder

The rest of this paper is organized into five sections: Section 2 provides the literature survey. Section 3 presents the proposed compression technique. Section 4 shows the experimental results. Section 5 concludes the paper with obtained results.

II. LITERATURE SURVEY

Dr.S.S.Pandey et al. discussed about the image compression techniques with DCT and quantization method for reducing the blocking artifacts in reconstruction. Several images were tested with this technique to obtain better PSNR of the reconstructed image [3]. XiHong Zhou studied about the DCT-based image compression quality by two evaluation methods: subjective and objective evaluations by computing PSNR and MSE of different images [4]. James S. Walker studied the image compression based on wavelets. He conducted experiments on compression techniques such as WDR, ASWDR and SPIHT to measure their efficiency [5].

Magnus Erik et al. studied the particle swarm optimization and proposed its simplified variant. They presented an easy technique for efficiently tuning the behavioral parameters and found that the simplified variant offered a small improvement in some cases [6]. Prabhjeet kaur et al. proposed a new compression technique by integrating the Particle swarm optimization (PSO) and Genetic algorithm (GA) based compression in wavelet domain. This technique is used for reducing the blocking artifacts in images [7]. Shet Reshma Prakash et al. made a review on Meta heuristic optimization algorithms which are used for image compression. They provided an insight of optimization techniques including Ant Colony Optimization (ACO) algorithm, Harfmony Search Algorithm (HSA), Artificial Bee Colony (ABC) algorithm, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) [8]. Aashish Kumar Bohre et al. introduced the Butterfly Particle Swarm Optimization technique with sensitivity and probability parameters. Experiments were conducted on various Benchmark functions. The results showed good convergence rate and more optimum output [9].

Thaneshwar Kumar et al. proposed a compression scheme based on hybrid DWT, DCT and Huffman coding techniques. Experiments were performed on several DICOM medical images and observed that the proposed method gives better quality for medical images [10]. S.Anitha et al. provided the practical ways of exploring arithmetic coding with Huffman coding techniques [11]. Rahul Shukla et al. studied the compression of Images using DCT and Huffman encoding techniques. The results were tabulated based on histogram information and image segmentation [12].

III. PROPOSED COMPRESSION TECHNIQUE

In this paper, the proposed compression technique contains DCT followed by BPSO for thresholding. Fig.2 shows the block diagram of the proposed compression method.

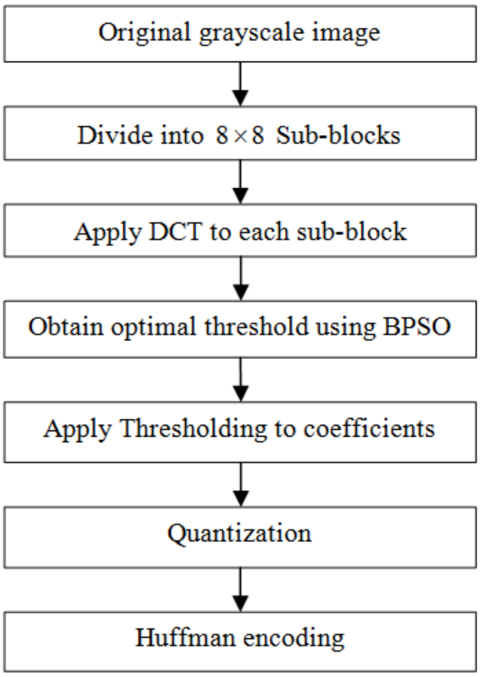


Fig.2. Block diagram of the proposed compression method

It consists of four stages as follows:
1) Transformation using DCT
2) Selection of number of non-zero coefficients using BPSO
3) Quantization
4) Huffman encoding

3.1 Discrete Cosine Transformation

Discrete Cosine Transform is the most widely used transform in digital signal processing systems. DCT can be applied to an image in sub-blocks of size $b \times b$ which could be 8×8, 16×16, 32×32 etc. Each sub-block will undergo DCT transformation. In our experiments, 8×8 sub-blocks are utilized. The forward and inverse equations of DCT are [13]:

$$F(m,n) = \frac{2}{\sqrt{MN}} C(m)C(n) \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} f(i,j)\cos\left[\frac{m\pi(2i+1)}{2M}\right]\cos\left[\frac{n\pi(2j+1)}{2N}\right] \quad (1)$$

$$f'(i,j)=\frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}C(m)C(n)\,F(m,n)\cos\left[\frac{m\pi(2i+1)}{2M}\right]\cos\left[\frac{n\pi(2i+1)}{2N}\right] \quad (2)$$

where

$$C(m),C(n)=\begin{cases}\frac{1}{\sqrt{2}}, & m,n=0\\ 1, & otherwise\end{cases} \quad (3)$$

$f(i,j)\rightarrow$ original image
$i,j \rightarrow$ discrete frequency variables
$MN \rightarrow$ dimension of image such as 8×8
$F(x,y)\rightarrow$ resultant image of forward DCT
$f'(x,y)\rightarrow$ resultant image of inverse DCT

The basic theory of DCT implies that after applying transformation to a matrix, the energy of the image is mostly positioned at upper-left corner of the output matrix while low energy coefficients are kept at lower-right part. This means that coefficients are allotted such that frequency of coefficients increases from upper-left to lower-right corner of the matrix. In Fig.3, $f(0,0)$ is the lowest frequency called DC coefficient and the remaining $f(0,1),...,f(7,7)$ with higher frequencies are called AC coefficients.

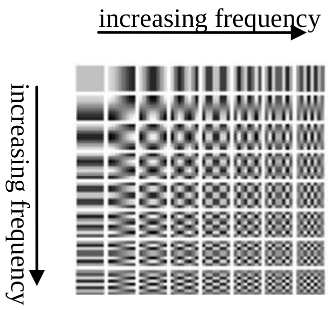


Fig.3. Visualization of 2D DCT Basis Functions

After applying DCT, the coefficients of 8×8 sub-blocks are arranged linearly from higher energy towards lower energy which is denoted as $d(n)$ where $1\le n\le b^2$ and thus $d(n)$ contains a total of b^2 elements.

3.2 Thresholding

Thresholding is a form of quantization where psycho-visual redundancy is removed. The relatively insignificant AC coefficients in the image are set to zero. To perform this, the number of coefficients (threshold) should be determined. In this work, the Butterfly-Particle Swarm Optimization (BPSO) is used for selecting the optimum threshold. The BPSO is described in next section. The obtained threshold is denoted by n_c. Once this value is obtained, it is used for all sub-blocks. The equation to perform thresholding is as follows:

$$\forall \quad d(n)\in sub-block \qquad d(n)=\begin{cases}d(n), & n\le n_c\\ 0, & n>n_c\end{cases} \quad (4)$$

Thus, the low energy coefficients are suppressed to zero. Now, the non-zero elements of $d(n)$ are re-arranged at their respective intensity positions to form the matrix $t(i,j)$. This step is done for all sub-blocks. The matrix $t(i,j)$ is the output matrix after thresholding.

3.3 Butterfly Particle Swarm Optimization

The novel idea of using BPSO for finding threshold is used in this research paper. BPSO is one of the optimization techniques which explores and acquires the ideas of the nature. The parameters used in BPSO are presented in this section. The size of butterfly swarm (number of particles) is denoted by N. Each particle is representing the number of coefficients. For example, if 8×8 DCT is used then it is tedious to search all 64 coefficients at a time to get optimal value. Instead, some N coefficients are selected randomly and searched. This search continues for a maximum number of iterations or until required compression ratio is reached. Let t denotes the iteration count and t_{max} denotes the maximum number of iterations. The position and velocity for particle k is represented by vectors x_k and v_k respectively. The position denotes the n^{th} coefficient value and velocity denotes the units for moving from one coefficient to another in following iterations to continue the searching process. Moreover, each particle k will maintain its personal best position at t denoted by $Lbest$. The overall best position achieved from whole swarm at t is $Gbest$. The velocity and position update equations with respect to iterations are given by:

$$v_i(t+1)=w(t).v_i(t)+s(t).(1-p(t)).r_1.c_1.(Lbest-x_i(t)) + p(t).r_2.c_2.(Gbest-x_i(t)) \quad (5)$$

$$w(t)=\frac{t_{max}-t}{t_{max}} \quad (6)$$

where $r_1,r_2\rightarrow$ uniformly distributed random numbers over [0,1]
$c_1,c_2\rightarrow$ cognitive and social acceleration rates
$w\rightarrow$ inertia weight

$$x_i(t+1)=x_i(t)+\alpha(t).v_i(t+1) \quad , \quad 1\le i\le N \quad (7)$$

where $\alpha(t)=rand.p(t)$, $rand\in[0,1]$ (8)
$\alpha(t)\rightarrow$ probability coefficient

The fitness function used in this work for selecting optimal threshold is as follows:

$$Fitness = a \times entropy + \frac{b}{PSNR} \quad (9)$$

The parameters a and b are adjustable numbers provided to modify compression gain and distortion as per user requirements. For example, if compression is the goal then a should be increased and b should be decreased. The entropy denotes amount of information carried in bits per pixel, while the PSNR denotes the quality of the reconstructed image. Along with these parameters, two additional parameters make the BPSO better from classic PSO technique. They are sensitivity S and probability P.

Fig. 4 shows search process of BPSO by considering an initial search space for Lena image with nodes $N1$ to $N8$ representing number of coefficients and the active region where optimal threshold is present. The region is selected based on sensitivity and probability.

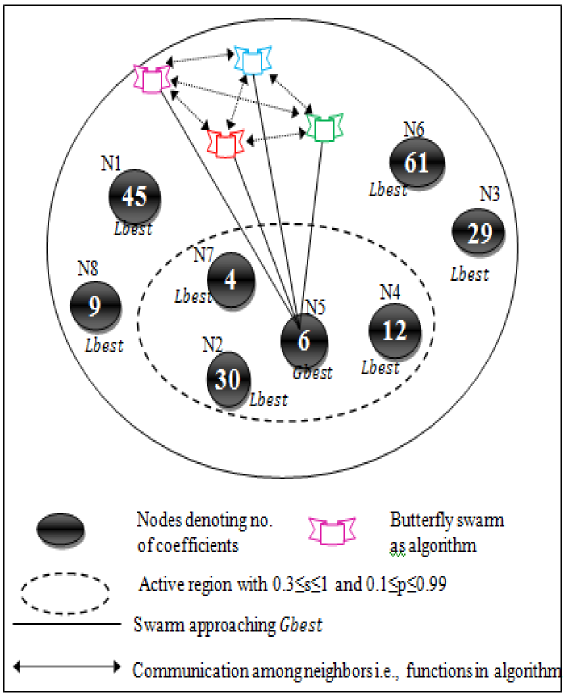


Fig.4. Butterfly Particle swarm optimization representation

The sensitivity tells how close the algorithm to find the optimum threshold. The probability says whether the selected nodes are nearer to the optimal node. The equations of sensitivity and probability are given by:

$$s(t) = e^{-\left(\frac{t_{\max}-t}{t_{\max}}\right)} \quad (10)$$

where $t_{\max} \rightarrow$ maximum number of iterations

$t \rightarrow$ current iteration

$$p(t) = \frac{Fit_{Gbest}}{\sum(Fit_{Lbest})} \quad (11)$$

where $Fit_{Lbest} \rightarrow$ Fitness of local best solutions

$Fit_{Gbest} \rightarrow$ Fitness of global best solution

When the sensitivity and probability of finding optimum threshold is higher in a particular region, it is called active region. The region with less probability and sensitivity is called inactive region. Hence, parameters S and P are used to decide moving direction towards active regions of the search space for consecutive iterations. By doing so, optimal solution is obtained quickly and efficiently. The boundary conditions for the active region are given by:

$$s(t) = \begin{cases} active, & 0.3 \le s(t) \le 1.0 \\ inactive, & otherwise \end{cases} \quad (12)$$

$$p(t) = \begin{cases} active, & 0.1 \le p(t) \le 0.99 \\ inactive, & otherwise \end{cases} \quad (13)$$

3.4 Quantization

Quantization is defined as the process of reducing the number of bits required to store an image [14]. It is a lossy compression technique. In DCT, quantization is done by dividing the given matrix by standard quantization matrix to suppress the AC coefficients. In this work, most of the high frequency coefficients with less information are rounded to zero in thresholding step. The remaining coefficients are floating point numbers and so occupy more bytes for representation. To reduce bit rate, these coefficients are rounded to integer values. A constant Q is used for the purpose of quantization. The quantization levels used here are 4, 8, 16 and 32. The equations for quantization and de-quantization are as follows:

$$Q = \frac{2^8}{q} \quad (14)$$

$$Q(i,j) = round\left(\frac{t(i,j)}{Q}\right) \quad (15)$$

$$Q'(i,j) = Q(i,j) * Q \quad (16)$$

3.5 Huffman Encoding

The Huffman encoding gives a data representation with the smallest possible number of code symbols for single source symbol. Here, the symbols denote coefficient values. The Huffman code construction is done in two steps:

3.5.1 Source reductions

Create a list of source reductions by arranging the probabilities of symbols and then combine the symbols of lowest probability to form new symbols recursively .This step is called as Huffman tree.

3.5.2 Code assignment

When only two symbols remain, start to assign bits (from 2 bits to maximum required) for all symbols so far discovered.

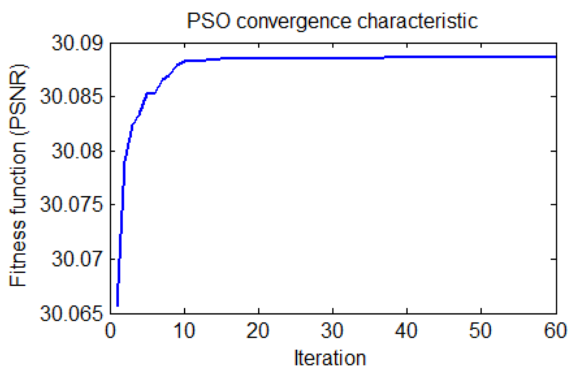
3.6 Proposed Encoding Algorithm

1: Get the input image $f(i,j)$ of size 512×512
2: Divide $f(i,j)$ into 64 8×8 blocks
3: **for** k=1 to 64 do
  $F(i,j) = DCT(f(i,j))$
end
4: Set $N=8$, $t_{\max}=60$, $t=1$
5: Initialize $x_i(0) = x_{\min} + rand().(x_{\max} - x_{\min})$
  where $[x_{\min}, x_{\max}] = [1, b^2]$ // $x_i \rightarrow$ position
6: Initialize $v_i(0) = v_{\min} + rand().(v_{\max} - v_{\min})$
  where $[v_{\min}, v_{\max}] = [-2,2]$ // $v_i \rightarrow$ velocity
7: Let $c_1 = c_2 = 2$ and $w(0) = 1$ where $0.8 \le w \le 1.2$
8: Evaluate fitness for each particle // by Eq. (9)
9: Find *Pbest* of each particle and *Gbest*
10: **while** $t < t_{\max}$ **do**
11: Calculate $s(t)$ and $p(t)$
  // using Eq. (1) and (3) respectively
12: **if** $(s(t) \,\&\&\, p(t)) == active$ // refer Eq. (12), (13)
  move towards active region
 else
  goto step 13
 end if
13: Update position, inertia and velocity vectors
14: Evaluate fitness and update *Pbest* , *Gbest*
15: **if** $Fitness(Gbest_t) \ge Fitness(Gbest_{t-1})$
  $Gbest = Gbest_t$
 else
  $Gbest = Gbest_{t-1}$
 end if
16: **end while**
17: Let $n_c = Gbest$
18: **for** k=1 to 64 **do**
  $t(i,j) = thresholding(F(i,j), n_c)$
 // perform thresholding function using Eq. (4)
 end
19: Compute Q value by Eq. (8)
20: $Q'(i,j) = quntization(t(i,j))$
 // perform quantization function using Eq. (16)
21: $hcode = huffmanenco(Q'(i,j))$
 // perform huffman encoding
22: Save $hcode$, Q and transmit to decoder

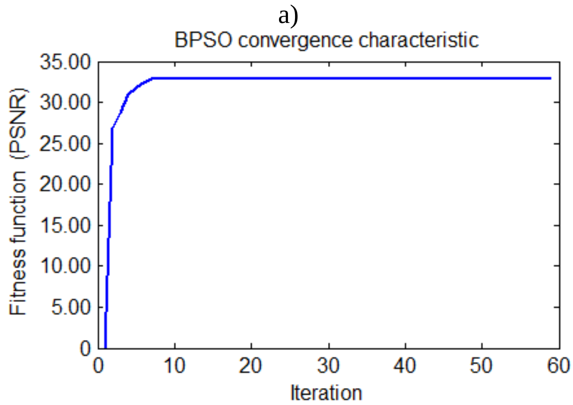
The encoder has compressed the original image. At the decoder stage, the reverse process takes place to get back the original image.

IV. EXPERIMENTAL RESULTS

Experiments are conducted on standard test images and MRI images. The implementation was done using MATLAB. The images taken as input are gray scale images of size 512×512. Standard test images including Lena, Woman, Lake, Walk bridge and Camera man are taken for experimentation. The results are compared with existing compression technique which uses DWT (Discrete Wavelet Transform) with PSO [15]. Fig.5 shows the convergence characteristic of both PSO and BPSO for Lena image where threshold is obtained at an early stage of iteration in BPSO.



a)



b)

Fig.5. Graphs showing convergence characteristic: a) PSO b) BPSO

From the graph, it is understood that BPSO effectively finds the threshold value at seventh iteration itself. It takes nearly 40 iterations for PSO to find the threshold value. The quality of the reconstructed image is evaluated using the formula [15]:

$$PSNR = 10 * \log\left(\frac{255^2}{MSE}\right) \tag{17}$$

where $$MSE = \frac{1}{MN}\sum_{i=0}^{M}\sum_{j=0}^{N}(I(i,j) - J(i,j))^2 \tag{18}$$

$PSNR \rightarrow$ Peak Signal to Noise Ratio
$MSE \rightarrow$ Mean Square Error

The compression ratio (CR) and bits per pixel (bpp) are found by using the equations:

$$CR = \frac{Uncompressed\ size\ in\ bits}{Compressed\ size\ in\ bits} \qquad (19)$$

$$bpp = \frac{Size\ of\ compressed\ image\ in\ bits}{Total\ number\ of\ pixels\ in\ image} \qquad (20)$$

Fig.6 shows the performance of the proposed technique in terms of Compression Ratio (CR) at *Q=32*. The proposed compression technique gives higher CR than the existing method. The thresholding and quantization discussed in this work greatly reduce the total bits needed for transmission and hence the higher compression ratio is achieved.

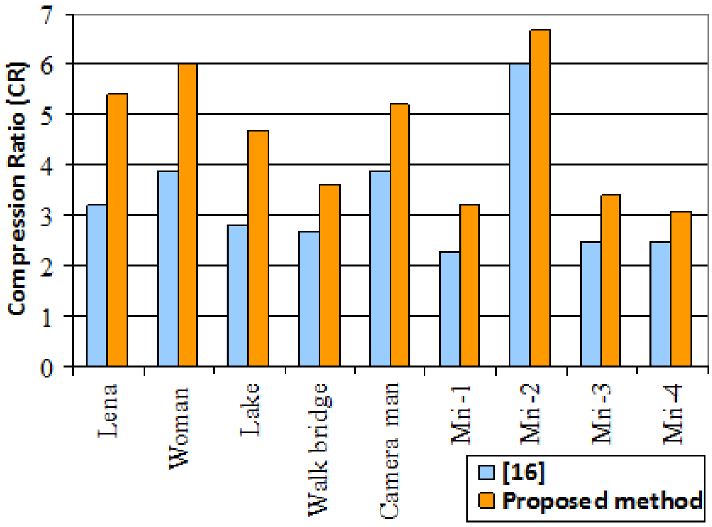
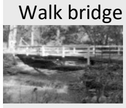


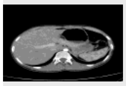
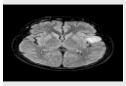
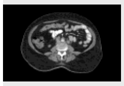
Fig.6. Bar chart showing CR of various images for existing and proposed techniques

The TABLE 1 shows the performance of proposed technique in terms of PSNR and bpp at various quantization levels. The quality of images increases with an increase in *Q* value. Beyond the level 32, PSNR could have better values with reduced CR but size of the reconstructed image increases .So, depending upon user's need, results could be obtained. On an average, the existing method (DWT+PSO) provides the PSNR 25.9655 for Q=4 and 30.2316 for Q=32. The average PSNR achieved with the proposed method (DCT+BPSO) is 27.9773 for Q=4 and 32.5593 for Q=32. By comparing the obtained results, it is evident that proposed method outperforms existing technique.

Fig. 7 shows the GUI (Graphical User Interface) model of the proposed compression technique.

TABLE 1. Results of proposed technique in comparison with existing technique at various quantization levels

Test Images		***Existing method [15]***				***Proposed Method***			
		q=4	**q=8**	**q=16**	**q=32**	**q=4**	**q=8**	**q=16**	**q=32**
Lena 	**PSNR** **bpp**	27.4036 1.3139	29.1953 1.5716	29.8869 1.9674	30.0968 2.5361	30.3944 1.1550	31.7598 1.2346	32.3108 1.3399	32.5221 1.4820
Woman 	**PSNR** **bpp**	28.7806 1.2118	31.8383 1.4010	33.2702 1.6875	33.7370 2.0395	33.3623 1.1148	35.2579 1.1682	36.1606 1.2407	36.4688 1.3195
Lake 	**PSNR** **bpp**	25.0875 1.3953	26.2418 1.7565	26.5853 2.2534	26.6827 2.8482	28.6709 1.2164	30.3416 1.3473	31.0407 1.5297	31.2538 1.7162
Walk bridge 	**PSNR** **bpp**	23.9088 1.4486	25.2495 1.8581	25.6679 2.4066	25.7804 2.9893	26.6416 1.2801	29.7468 1.5519	31.0057 1.8833	31.3791 2.2013
Camera man									

	PSNR	25.7933	27.5290	28.3013	28.4988	32.1543	35.0893	36.6057	37.2131
	bpp	1.2623	1.4418	1.6915	2.0415	1.1549	1.2489	1.3676	1.5315
Mri-1 	PSNR	25.4998	26.3394	26.5870	26.6539	28.4069	28.8496	28.9318	28.9533
	bpp	2.4077	2.6465	3.0093	3.4969	2.3090	2.4013	2.4709	2.5264
Mri-2 	PSNR	23.3638	23.5956	23.6494	23.6668	30.2825	30.8085	30.9284	30.9669
	bpp	1.1023	1.1932	1.2752	1.3330	1.0865	1.1269	1.1593	1.1883
Mri-3 	PSNR	27.7517	29.0306	29.3970	29.4987	31.3495	32.2830	32.5152	32.5681
	bpp	2.0047	2.2956	2.7258	3.1436	2.1388	2.2290	2.3037	2.3656
Mri-4 	PSNR	26.1004	26.8250	27.1131	27.1811	30.8226	31.4608	31.6684	31.7092
	bpp	2.1142	2.3590	2.7591	3.1936	2.3689	2.4574	2.5405	2.6059

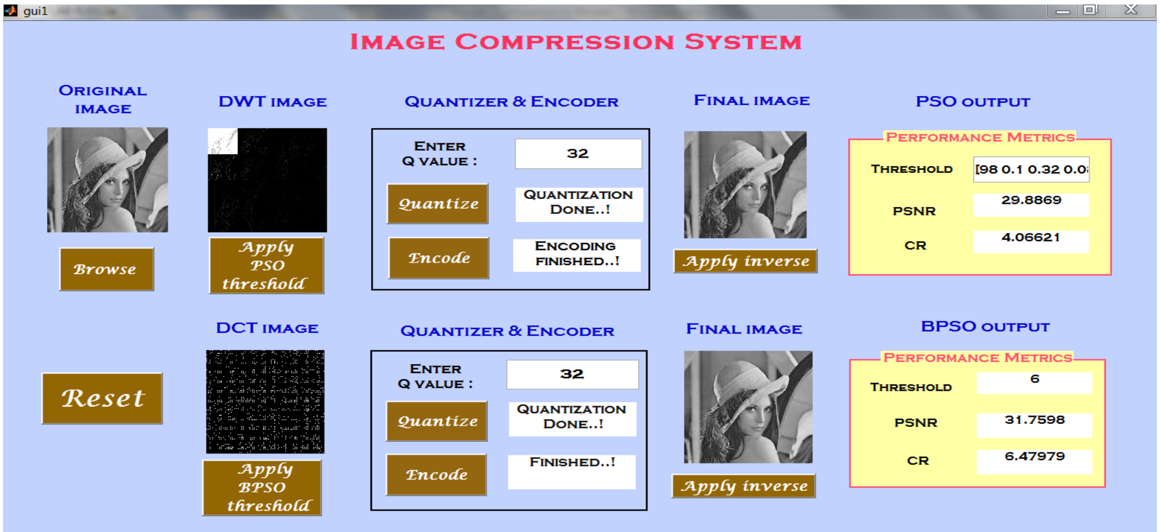


Fig.7. Screenshot of proposed system in GUI

V. CONCLUSION

The proposed image compression technique which uses DCT and BPSO for threshold selection was implemented in MATLAB. The experiments show higher performance in terms of both compression and PSNR when compared to existing method using PSO technique. The threshold obtained using BPSO provided good compression rate. The better convergence rate of BPSO induced the efficiency of encoder. Hence BPSO can be adopted for optimization problems in image compression systems to get better results along with DCT.

REFERENCES

[1] Hanna ZainEldin, Mostafa A. Elhosseini and Hesham A Ali, Image compression algorithms in wireless multimedia sensor networks: A survey, *Ain Shams Engineering Journal, vol. 6* pp. 481-490, December 2014.

[2] Ching-Hung Yuen and Kwok-Wo Wong, A chaos-based joint image compression and encryption scheme using DCT and SHA-1, *Applied Soft Computing, vol.11*, pp. 5092-5098, December 2011.

[3] Dr.S.S.Pandey, Manu Pratap Singh & Vikas Pandey, Block wise image compression & Reduced Blocks Artifacts Using Discrete Cosine Transform, *International Journal of Scientific and Research Publications, Volume 5, Issue 3*, March 2015.

[4] XiHongZHOU, Research on DCT -based Image Compression Quality, *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference*, 2011.

[5] Prabhjeet kaur, Er. Parminder Singh, Integrated Particle Swarm Optimization and Genetic Algorithm Based Compression forReduction of Blocking Artifacts, *International Journal of Computer Science*

and Information Technologies, Vol. 6 (5) , 2015, 4520-4527.

[6] Magnus Erik Hvass Pedersen, Andrew John Chipper, Simplifying Particle Swarm Optimization, Published in *Applied Soft Computing*, 2009.

[7] Shet Reshma Prakash, Vrinda Shetty, Review on optimization techniques used for image compression, *International Journal of Research in Engineering and Technology*, eISSN: 2319-1163 | pISSN: 2321-7308, *Volume: 04 Issue: 02* | Feb-2015, Available @ http://www.ijret.org.

[8] A.K. Bohre, G. Agnihotri, and M. Dubey, Hybrid butterfly based particle swarm optimization foroptimization problems, First International Conference on *Networks & Soft Computing (ICNSC)*, 2014, pp.172-177, 19-20 Aug. 2014, doi: 10.1109/CNSC.2014.6906650.

[9] Thaneshwar Kumar, Dr. Ramesh Kumar, Medical Image Compression Using Hybrid Techniques of DWT, DCT and Huffman Coding, *Vol. 3, Issue 2,* February 2015, *IJIREEICE* DOI 10.17148/IJIREEICE.2015.3214 54.

[10] Anitha. S, lossless image compression and decompression using huffman coding , *International Research Journal of Engineering and Technology (IRJET)* e-ISSN: 2395-0056 *Volume: 02 Issue: 01* | Apr-2015 www.irjet.net p-ISSN: 2395-0072.

[11] James S. Walker, Wavelet-based Image Compression, Sub-chapter of CRC Press book: *Transforms and Data Compression.*

[12] Rahul Shukla and Narender Kumar Gupta, Image Compression through DCT and Huffman Coding Technique, *International Journal of Current Engineering and Technology,* 2015.

[13] Binh P. Nguyen, Chee-Kong Chui, Sim-Heng Ong and Stephen Chang, An efficient compression scheme for 4-D medical images using hierarchial vector quantization and motion compensation, *Computers in Biology and Medicine, vol. 41*, pp. 843-856, September 2011.

[14] Shogo Okamoto, Lossy Data Compression of Vibrotactile Material-Like textures, *IEEE Transactions on Haptics, vol. 6* pp. 69-80, March 2013.

[15] Kaveh Ahmadi, Ezzhatollah Salari, An efficient compression scheme based on adaptive thresholding in wavelet domain using particle swarm optimization, Elsevier, 2015.