

# Effective lossy and lossless color image compression with Multilayer Perceptron

Dr. PL. Chithra \*, A. Christoper Tamilmathi

Department of Computer science, University of Madras, Chennai, India

\*Email: [chitra.cs@unom.ac.in](mailto:chitra.cs@unom.ac.in)

## Abstract

This paper presents the effective lossy and lossless color image compression algorithm with Multilayer perceptron. The parallel structure of neural network and the concept of image compression combined to yield a better reconstructed image with constant bit rate and less computation complexity. Original color image component has been divided into 8x8 blocks. The discrete cosine transform (DCT) applied on each block for lossy compression or discrete wavelet transform (DWT) applied for lossless image compression. The output coefficient values have been normalized by using mod function. These normalized vectors have been passed to Multilayer Perceptron (MLP). This proposed method implements the Back propagation neural network (BPNN) which is suitable for compression process with less convergence time. Performance of the proposed compression work is evaluated based on three ways. First one compared the performance of lossy and lossless compression with BPNN. Second one, evaluated based on different sized hidden layers and proved that increased neurons in hidden layer has been preserved the brightness of an image. Third, the evaluation based on three different types of activation function and the result shows that each function has its own merit. Proposed algorithm has been competed with existing JPEG color compression algorithm based on PSNR measurement. Resultant value denotes that the proposed method well performed to produce the better reconstructed image with PSNR value approximately increased by 21.62%.

**Keywords:** Activation function; Back propagation neural network; Discrete cosine transform; Error; Hidden layer; JPEG compression.

## 1. Introduction

Nowadays communication medium rapidly transformed with image transformation. So the image compression has taken the essential part to reduce not only the storage and band width of the transmission and also the time and cost. Image compression has been done in two ways, lossy and lossless compression. JPEG compression method is the one of the most significant lossy compression algorithm. Another one is DWT for lossless image compression. This proposed work deals with both JPEG and DWT compression combined with multilayer perceptron. BPNN is one of the Multilayer perceptron algorithms. Combination of BPNN and JPEG compression algorithm eliminates the need of quantization table and entropy symbol table and produces the constant bit rate compression. The Training algorithm and back propagation neural network is used to increase the performance and to decrease the convergence time and provide high compression ratio with low distortion [1]. Effectively compress a wide range of novel images using back propagation. The networks operate and are trained on residual image blocks [2]. Haar wavelet transform and discrete cosine transform are considered and a neural network is trained to relate the x-ray image which controls compression method and their optimum compression ratio [3]. Back propagation neural network algorithm helps to increase the performance of the system and to decrease the convergence time for the training of the neural network [4]. Fast BP results such as compression ratio (CR) and peak signal to noise ratio (PSNR) are computed and compared with BP results. From the results, we noticed that the use of FBP improve the BPNN training by reducing

the convergence time of image compression learning process [5]. Neural network is well suited for real time systems because of their fast response and computational times which are due to their parallel architecture [6]. If a network gets trained up successfully with a particular activation function, then there is a high probability that other activation function will also lead to proper training of neural network [7].

The concept of image compression using ANN and DWT possess the advantages of simple computations, fault tolerance, parallel processing, robust with respect to error transmission in the communication media which has made a breakthrough in supervised learning of layered neural network [8]. Keeping the quality of the image considerably same even after transmitting via noisy channel in their BPNN algorithm. Thus the data transmission can be done with less band width, power and storage space [9]. JPEG image compression using FPGA with artificial neural networks gives more compression ratio when compared to existing systems [10]. The implement of ANN with CSD based multiplier is thus proven to be one of the effective processes for image compression and decompression [11]. Steps of back propagation algorithms and manual calculations have been displayed [12]. In image compression technique convergence time also play main role for quality of image and the back propagation neural network and Levenberg - marquardt algorithms estimating a connection counted in which image compression and convergence time have been improved [13]. The bipolar technique is proposed and implemented for image compression and obtained the better results as compared to principal component analysis (PCA technique). It is observed that the Bipolar & LM algorithm suits the

best for image compression processing applications [14]. Hyperbolic tangent function has better performance than sigmoid function as the new back propagation network's activation function for image compression [15].

In this paper the effective color image compression has been done using multilayer perceptron (BPNN) to reduce the spaces needed for storing quantization table and entropy symbol table, low computation complexity, constant bitrate with good quality reconstructed image. The rest of the paper is organized as follows. Section II explains the proposed work. Section III discusses the experimental results. Concluding the proposed work has been given in Section IV.

## 2. Proposed work

### 2.1. Image compression

The general architecture of proposed compression work shown in figure 1(a). The RGB format of an input image converted to the YCBCR format then decomposed into the three isolated independent components. Each component further divided into the 8x8 blocks of coefficients. For lossy compression, discrete cosine transform has been applied on every block in each component to separate the high frequency coefficients from low frequency coefficients. DWT has been applied to extract the approximation coefficients from detailed coefficients to implement lossless image compression. This proposed method implements the haar wavelet with 4 decomposition level. The output frequency coefficients have been normalized using mod function. The BPNN can only accept the vector format of data to train and testing process [8]. The 8 x 8 blocks of normalized data converted into the column vector using zigzag scanning process. Algorithm of proposed image compression process has been shown in below

Algorithm: 1

- Step 1: Read an input RGB image
- Step 2: Convert the RGB into YCBCR image format
- Step 3: YCBCR image convert into Y, CB and CR component.
- Step 4: For each Component
  - Step 4.1: Divided in to 8x8 blocks
  - Step 4.2: Apply DCT on each block for lossy compression or apply DWT (haar wavelet with 4 decomposition level) for lossless compression.
  - Step 4.3: Normalization using mod function
  - Step 4.4: Zigzag scanning process
  - Step 4.5: Column vector move to the BPNN.
- Step 5: Reconstruction process

This proposed JPEG and DWT compression algorithm similar to the standard JPEG algorithm except the process of quantization and entropy coding. This work saves the memory spaces which are needed to store the quantization table of each component and built in entropy reference table. This novel compression algorithm utilized the performance of BPNN to improve the constant bit rate transferring with low computational complexity and produced a better reconstructed image.

### 2.2. Back propagation network

Back propagation network is one of the best training algorithms for multi-layer forward network model using extend gradient-descent based delta-learning rule. This algorithm is one of the most suitable architecture for compression process. The aim of this algorithm is to train the net to achieve the target value by minimizing the delta (error) value. The proposed BPNN architecture shown in figure 1(b). It consists of input layer, hidden layer and output layer.

The back propagation training algorithm consists of four steps as below [13].

1. Initialization of weights: Small distributed random weights have been initialized.
2. Feed forward process: Each input neuron ( $X_i$ ) has been received the input value and transferred to the hidden layer ( $Z_i$ ). Activation function (transfer function) has been computed in each hidden layer using net weight value.
3. Back propagation of errors: Each output value  $Y_k$  related to the target value  $t_k$  to find the error value. Based on the error the factor  $\delta_k$  ( $k = 1, \dots, m$ ) has been computed and distributed back to all neurons in the previous hidden layer.
4. Updating the weights and biases: The weight and biases have been renewed using  $\delta$  factor.

Proposed BPNN architecture has 64 input neurons (nodes) and 16 hidden neurons with 8000 epochs shown in figure 1(b). Proposed multilayer perceptron split into transmitter (encoder) and receiver (decoder). The column vector of the normalized coefficients which have been calculated from the image compression process is passed to the input neurons of the BPNN. In the hidden layer, weights have been initialized with random values and multiplied by the vectors to calculate the net weight. The activation function has been calculated in the hidden layer by using the net weight value. The back propagation process followed when the bit stream from the hidden layer have not been reached the target compressed bit stream. This back propagation process continued until the hidden layer produce the targeted bit stream or reach the epoch rate. Back propagation method has been worked as the quantization process in the compression algorithm. Output stream (compressed image) of hidden layer passed through the channel. Receiver end collects the bit stream from the channel and moved to the input of hidden layer to calculate the output value. To achieve the target value, the  $\delta$  factor value normally distributed to the previous hidden units. This process continued until the iteration reached the maximum fixed value. The output stream of the hidden layer in the receiver side passed as the input to the inverse process (decoding) of compression algorithm for reconstruct the original image. Algorithm of proposed BPNN has been explained below

Algorithm: 2

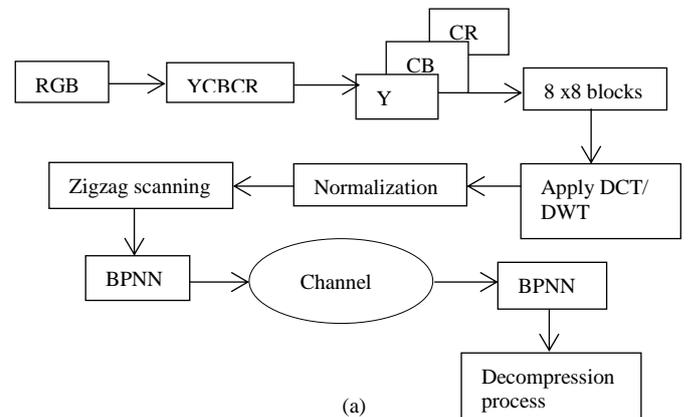
- Step 1: Initialize the small random weights.
- Step 2: For each input column vector do step 3 to Step 5.
- Step 3: For  $i = 1, 2, 3, \dots, n$ . activation of input  $X_i$ .
- Step 4: For  $j = 1, 2, 3, \dots, l$ .

$$Z_{-inj} = V_{oj} + \sum_{i=1}^n X_i V_{ij}$$

- Step 5: For  $k = 1, 2, 3, \dots, m$

$$Y_{-ink} = W_{ok} + \sum_{j=1}^l Z_j W_{jk}$$

$$Y_k = f(Y_{-ink})$$



(a)

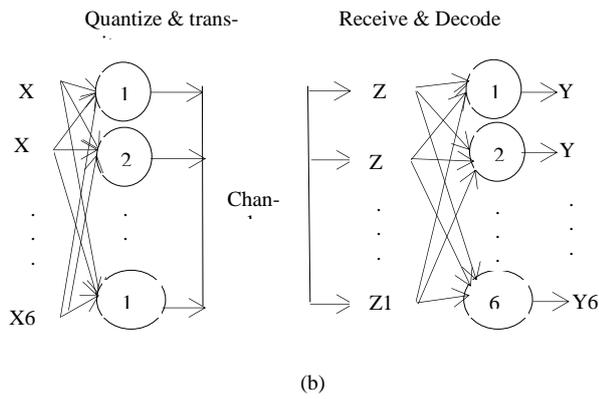


Fig. 1: (a) General architecture of proposed work. (b) Architecture proposed BPNN.

### 3. Experimental results

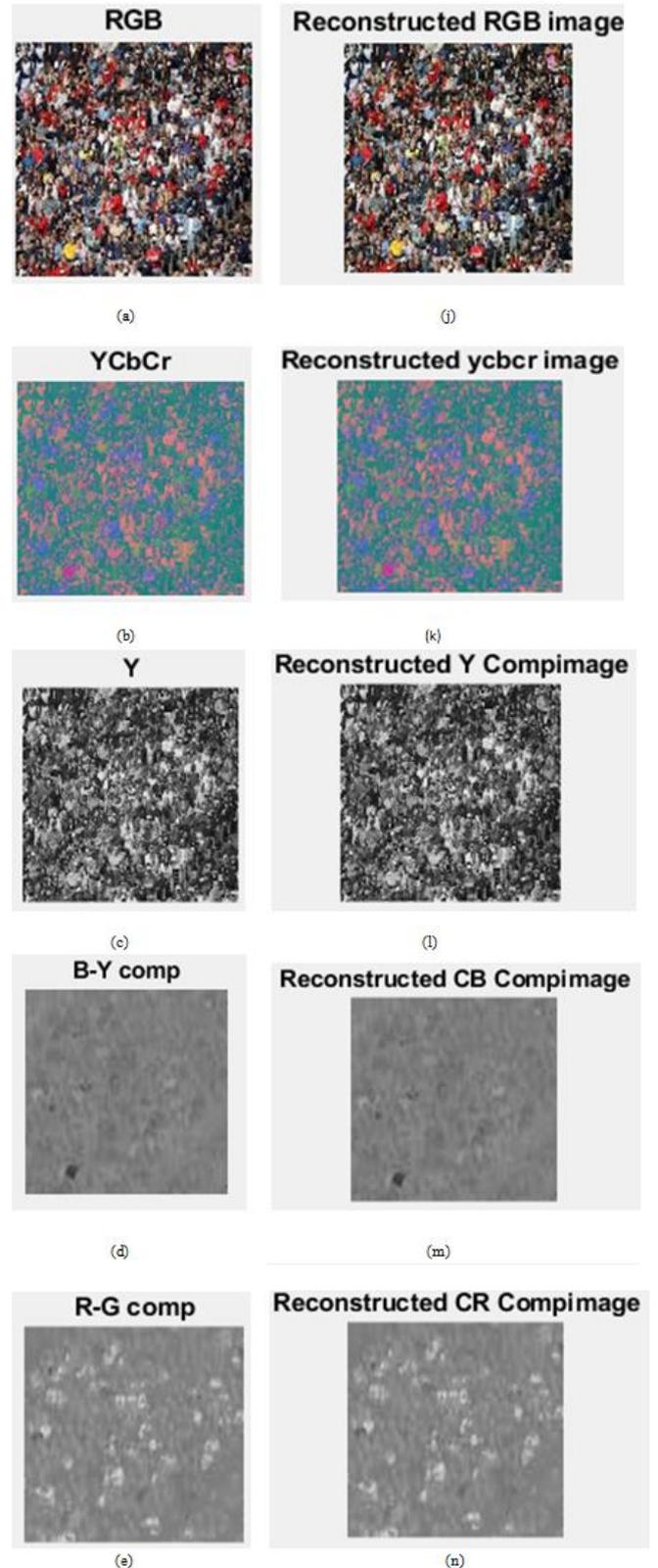
Performance of proposed work has been evaluated on different types of input color images (512 x 512) taken from image processing databases [17 and 18] shown in Fig. 2. This paper has been implemented using Matlab 2017b. The different psycho visual image samples are standard image (SI), more informative image (MII), less informative image (LII), low contrast image (LCI) and darker image (DI). The original RGB input image converted to the YCBCR image and then converted to the isolated independent luminance(Y), chrominance blue (CB) and chrominance red (CR) components. Each component has been divided into 8 x 8 blocks. In this proposed method each component consists of 4096 blocks. Transformation process has applied on each block to identify the low frequency from high frequency coefficients by using DCT for making better compression. For lossless compression, DWT has applied on each block to extract the approximation coefficients. Approximation coefficients are having the required information of an image. The transformed frequencies have been normalized using mod function and then converted into the column vector and passed to the input of BPNN. These are the forward process of the compression work. JPEG method implemented on one of the sample input image, the forward process as shown in figure 3(a) to 3(e).

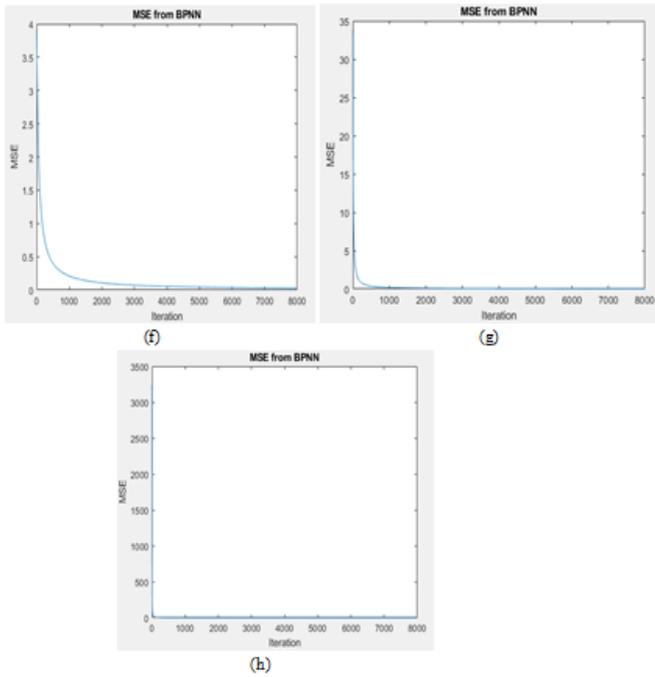


Fig. 2: Input RGB images (a).Standard image Lena, (b) More informative image, (c) Less informative image,(d) Low contrast image,(e) Darker image.

The reverse process can possible to make the reconstructed image as shown in figure 3(j) to 3(n). Entropy coding and quantization has not been implemented here but BPNN process in the forward compression method performed as the quantization process to compress

the normalized vector. Initially the compression process has been trained and evaluated using a hidden layer with 16 nodes, sigmoid activation function with 8000 epochs on MII has been shown in figure 3. Figure 3(f) shows the decreasing of mean square error (MSE) from highest value to lowest value for each epoch in Y component BPNN, (g) shows CB component BPNN and (h) shows CR component BPNN.





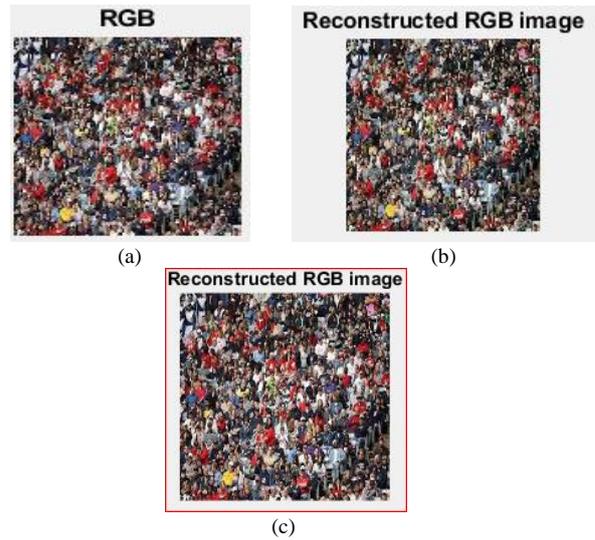
**Fig. 3:** Compression and decompression process.(a) Original RGB image (b)YCBCR image (c)Y component (d)CB component (e)CR component (f) Y-MSE (g) CB-MSE (h) CR-MSE (j) Reconstructed Y (k) Reconstructed CB (l) Reconstructed CR (m) Reconstructed YCBCR (n) Reconstructed Original image.

In this proposed work, the performance comparison of lossy (JPEG) and lossless (DWT) compression work with BPNN having 16 neurons and sigmoid activation function evaluated and tabulated in Table 1. Reconstructed images of JPEG and DWT methods are shown in figure 4. This figure illustrated that DWT (lossless) produces more replicated reconstructed image than the JPEG (lossy) due to the nature of the compression technique.

**Table 1.** Performance comparison of proposed lossy and lossless compression with BPNN.

Images		Hidden layer with 16 nodes (Sigmoid function)			
		PSNR (DWT)	PSNR (JPEG)	Exe. time(Sec) (DWT)	Exe. time(Sec) (JPEG)
Standard image(SI)	Y	Inf	45.12	720.35	548.27
	CB	Inf	45.05	715.08	544.19
	CR	Inf	45.22	714.61	548.66
More informative image(MII)	Y	Inf	45.15	728.84	546.30
	CB	Inf	45.08	712.52	551.61
	CR	Inf	45.15	716.03	548.77
Less informative image(LII)	Y	Inf	45.22	716.91	555.58
	CB	Inf	45.18	740.71	552.66
	CR	Inf	45.08	720.31	556.18
Low contrast image(LCI)	Y	Inf	45.08	704.87	556.73
	CB	Inf	45.08	706.26	556.91
	CR	Inf	45.12	698.09	559.21
Darker image(DI)	Y	Inf	45.08	698.39	552.80
	CB	Inf	45.08	1085.02	548.12
	CR	Inf	45.08	690.32	552.65

Our experimental results in Table 1 clearly shows that the proposed DWT reproduces the exact reconstructed image like original image without error than the proposed JPEG but it consumes approximately 26.45% more time than the proposed JPEG.



**Fig. 4:** Comparison of the reconstructed images from the proposed work. (a) Original RGB image. (b) From proposed JPEG (lossy) method. (c) From proposed DWT (lossless) method.

The following, BPNN evaluation methods are based on proposed JPEG compression technique. Proposed compression method has been evaluated based on different sized (number of neurons) hidden layers and types of activation function.

Performance evaluation has been done using hidden layer with 8 neurons and then hidden layer with 16 neurons to compress the five visually different RGB image. Measurement of mean square error (MSE), peak signal noise ratio (PSNR), structure similarity index (SSIM) and execution time have been evaluated and tabulated in Table 2 and Table 3. Each block has generated single bit as a compressed bit (target bit). Sample image contains 4096 blocks has produces 4096 bit stream after the compression process. So the proposed work generate constant compression ratio of all sample images. One of the samples of reconstructed SI image from different sized hidden layers using JPEG method is shown in figure 5. It clearly signifies that reconstructed SI image, from the layer with 16 neurons, constructed better quality image than the image from the layer with 8 neurons. The proposed work infers that when the number of neurons increases in a hidden layer, it yields the compressed image with good quality due to increase of training process and accuracy of target bit stream.



**Fig. 5:** Decompressed image from different sized hidden layers.(a) Original image (b) Hidden layer with 8 neurons (c) Hidden layer with 16 neurons.

**Table 2.** Metrics of reconstructed image from hidden layer with 8 neurons.

Images		Hidden layer with 8 nodes (Sigmoid function)			
		MSE	PSNR	SSIM	Exe. time(Sec)
Standard image(SI)	Y	2.0	45.12	0.9995	548.27
	CB	2.03	45.05	0.9990	<b>544.19</b>
	CR	<b>1.95</b>	<b>45.22</b>	0.9998	548.66
More informative image(MII)	Y	1.98	45.15	<b>0.9999</b>	546.30
	CB	2.01	45.08	0.9992	551.61
	CR	1.98	45.15	0.9993	548.77
	Y	<b>1.95</b>	<b>45.22</b>	0.9988	555.58

<b>Less in- formative image(LII)</b>	CB	1.96	45.18	0.9988	552.66
	CR	2.01	45.08	0.9989	556.18
<b>Low con- trast im- age(LCI)</b>	Y	2.01	45.08	0.9991	556.73
	CB	2.01	45.08	0.9990	556.91
<b>Darker im- age(DI)</b>	CR	2.0	45.12	0.9988	559.21
	Y	2.01	45.08	0.9990	552.80
	CB	2.01	45.08	0.9990	548.12
	CR	2.01	45.08	0.9991	552.65

Table 3. Metrics of reconstructed image from hidden layer with 16 neurons.

Images	Hidden layer with 16 node (Sigmoid function)				Exe. time(Sec)
	MSE	PSNR	SSIM		
<b>Standard image(SI)</b>	Y	2.0	45.12	0.9995	581.32
	CB	2.01	45.08	0.9988	585.61
<b>More in- formative image(MII)</b>	CR	2.0	45.12	0.999	580.42
	Y	2.0	45.12	<b>0.9999</b>	583.31
<b>Less in- formative image(LII)</b>	CB	2.0	45.12	0.9991	585.63
	CR	2.0	45.12	0.9993	582.71
<b>Low con- trast im- age(LCI)</b>	Y	<b>1.95</b>	<b>45.22</b>	0.9988	591.85
	CB	1.95	45.22	0.9986	584.05
<b>Darker im- age(DI)</b>	CR	2.0	45.12	0.9989	590.47
	Y	2.0	45.12	0.999	593.47
	CB	2.03	45.05	0.9988	590.50
	CR	1.98	45.15	0.999	591.14
	Y	2.03	45.05	0.999	588.17
	CB	2.0	45.12	0.999	<b>577.83</b>
	CR	2.0	45.12	0.9991	584.32

From Table 2 and Table 3 concluded both the layers produced low MSE and high PSNR value for Y component in LII. So both layer types have been preserved the brightness of LII. Proposed method produced the MII reconstructed image more perfect than the other images according to the structure and preserved the information of LII image.

Performance evaluation of proposed work according to activation function has been done by three different types namely, sigmoid function, hyperbolic tangent function and Elliot sigmoid function. Activation function defines the output of the layer depends on the given input or set of inputs [7].

- *Sigmoid activation function:*

Probability of inputs exists only between the range of 0 and 1, sigmoid is the perfect activation function for training the data. Output produces positive numbers between 0 and 1. Sigmoid function has been calculated using (1) given below.

Span:  $0 < y < 1$

$$y = \frac{1}{1 + exp^{-x}} \quad (1)$$

- *Hyperbolic tangent activation function:*

Output of hyperbolic function produces between the ranges -1 to 1. The advantage is that the negative input has been mapped near zero in the tanh graph. It is one of the most used activation function. Output has been calculated using (2) given below.

Span:  $-1 < y < 1$

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2)$$

- *Elliot sigmoid activation function:*

Elliot function as similar as sigmoid function without exponential term. Output produced the ranges 0 to 1. Elliot function computed using (3) shown below.

Span:  $0 < y < 1$

$$\sigma_e(x) = \frac{1}{1 + |x|}$$

It can be written as

$$\sigma_e(x) = \frac{0.5(x)}{1 + |x|} + 0.5 \quad (3)$$

These three activation function has applied on five different RGB images through the hidden layer with 16 neurons to form the reconstruct image. One of the samples reconstructed DI from the JPEG with different activation function has been shown in figure 6.

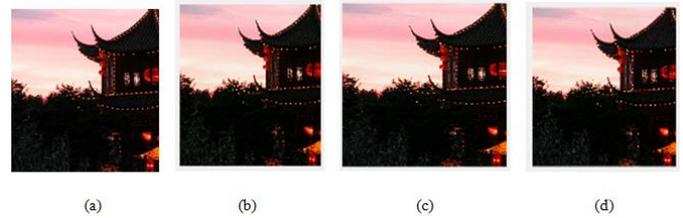


Fig. 6: Decompressed DI image from different transfer function. (a) Original DI image (b) Sigmoid function (c) Hyperbolic tangent function (d) Elliot sigmoid function.

Figure 6 shows the better quality reconstructed image has been produced by hyperbolic tangent activation function than the other two functions. The PSNR value of the five reconstructed images from different activation function have been determined and plotted in figure 7. The SSIM and execution time of reconstructed images have been calculated and plotted in figure 8 and figure 9.

PSNR from Transfer function

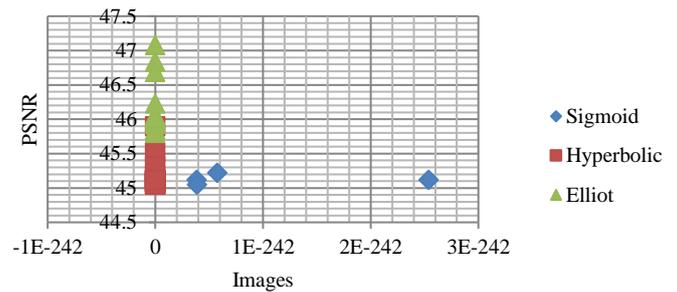


Fig. 7: PSNR value from different transfer functions

SSIM from Transfer function

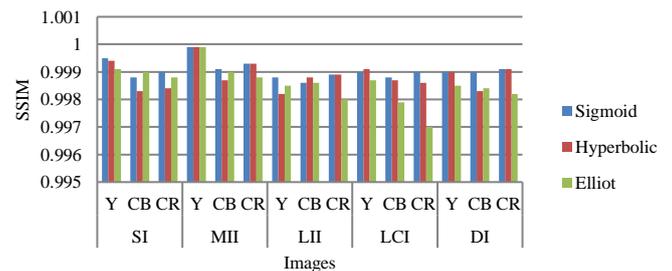


Fig. 8: SSIM produced from different activation function.

Execution time

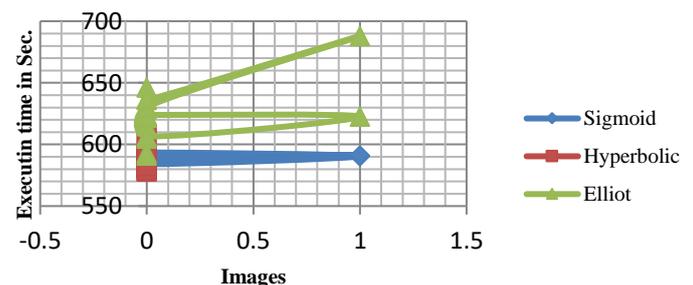


Fig. 9: Execution time of different activation function on input images.

From the figure 7 and figure 8, concluded that Elliot sigmoid function produced the reconstructed image with low MSE and high PSNR than the other two functions. Sigmoid function produced the reconstructed image with high SSIM value. Figure 9 shows the sigmoid and hyperbolic tangent function execute in same rate.

**Table 4.** Performance comparison between existing and proposed method.

Images	PSNR (dB)				
	JPEG (Existing method)	Proposed Method			Elliot sigmoid function
		Sigmoid function	Hyperbolic Tangent function	Elliot sigmoid function	
Standard image(SI)	Y	36.63	45.12	45.08	46.02
	CB	44.55	45.08	45.19	47.08
	CR	43.90	45.12	45.90	46.84
More in- formative image(MII)	Y	34.04	45.12	45.12	45.85
	CB	43.85	45.12	45.08	45.80
	CR	42.45	45.12	45.12	46.02
Less in- formative image(LII)	Y	43.55	45.22	45.52	45.98
	CB	45.42	45.22	45.05	46.69
	CR	42.77	45.12	45.08	45.97
Low con- trast im- age(LCI)	Y	38.31	45.12	45.12	46.06
	CB	46.37	45.05	45.05	45.93
	CR	44.87	45.15	45.32	46.02
Darker im- age(DI)	Y	37.39	45.05	45.05	45.98
	CB	43.27	45.12	45.12	46.02
	CR	40.40	45.12	45.12	46.23

Table 4 evaluates the PSNR values of the proposed method with existing JPEG color compression algorithm without using MLP. It signifies that proposed compression method out performs well to produce better reconstructed image with increased PSNR value than the existing JPEG compression. Elliot function has produced better PSNR value than the other two functions.

## 4. Conclusion

This paper presents the effective lossy and lossless image compression algorithm based on back propagation (MLP) algorithm. Proposed compression algorithm eliminates the need of quantization table, entropy coding and produced the constant bit rate compression. Suggested work performs well to produce better reconstructed image with low computation complexity. From the comparison table of proposed lossy and lossless method, illustrated that the DWT produces better reconstructed lossless image than the JPEG but it consumes approximately 26.45% more time than JPEG. Performance evaluation has been performed on different types of images and results show that when increases neurons in a hidden layer have preserves the brightness of an image. MSE, PSNR, SSIM and execution time have been computed from three different types of activation function and the graphical results shows that Elliot function has produced the better PSNR value, Sigmoid function has produced better structure similarity of reconstructed image as the original image, sigmoid and hyperbolic functions have same execution pace. PSNR value of the suggested method has been compared with existing JPEG color compression algorithm (without MLP). Resultant value denotes that the proposed method well performed to produce the better reconstructed image with PSNR value approximately increased by 21.62%. Final results infer that proposed method has produced better reconstructed image than the existing method. Among the three activation function Elliot is the suitable activation function for color image compression according to the PSNR value. For future work different compression algorithm could be performed with different neural network architecture.

## References

- [1] B. K. Patel, S. Agarwal, "Image compression techniques using artificial neural network", *International Journal of Advanced Research in Computer Engineering & Technology*, Vol. 2, No. 10, (Oct 2013).
- [2] G. Qiu, T. J Terrel, M. R. Varely, "Improved image compression using back propagation networks", *Neural Network Applications and Tools - IEEE*, DOI: 10.1109/NNAT. 1993. 586056, (1994), pp. 73 – 81.
- [3] K. Dilmililer, "Neural network implementation for medical image compression", *Journal of applied Mathematics*. (2013) pp. 1 – 8.
- [4] N. Jaiwal, "Image compression using back propagation neural network", *International Journal of Science Engineering and Research*, Vol. 3, No. 5, (2015), pp. 61 – 64.
- [5] N. Omaira, A. AL - Allaf, "Fast back propagation neural network algorithm for reducing convergence time of BPNN image compression", *Proceedings of the fifth International conference on IT & multimedia at ONITEN-IEEE*, (Nov 2011), pp. 14 -16.
- [6] P. K. Charles, H. Khan, Ch. R. Kumar, N. Nikitha, S. Roy, V. Harish, M. Swathi, "Artificial neural network based image compression using Levenberg-Marquart algorithm", *International Journal of Modern Engineering Research*, Vol. 1, No.2, pp. 482 - 489.
- [7] P. Sibi, S. A. Jones, P. Siddarth, "Analysis of different activation functions using back propagation neural networks", *Journal of theoretical and applied information technology*, Vol. 47, No. 3, pp. 1264 – 1268.
- [8] R. A. Vasmatkar, S. P. Biradar, P. B. Shivashankar, "Artificial intelligence used for image compression", *Journal of BIOINFO Computational mathematics*, Vol. 1, No. 1, (2011), pp. 05 – 10.
- [9] S. A. Ahamed, K. C. Shekarappa, "ANN implementation for image compression and decompression using back propagation techniques", *International Journal of Science and Research*, Vol. 3, No. 6, (2014), pp. 1848 – 1851.
- [10] S. A. K. Jilani, S. A. Sattar, "JPEG Image compression using FPGA with artificial neural networks", *International Journal of Engineering and Technology*, Vol. 2, No. 3, (2010), pp. 252 -257.
- [11] S. L. Pinjare, E. H. Kumar, "Implementation of Artificial neural network architecture for image compression using CSD multiplier", *Emerging Research in Computing, Information, Communication and Applications- Elsevier*, (Aug 2013), pp. 581 – 587.
- [12] S. N. Sivanandam, S. Sumathi, S. N. Deepa, "Introduction to neural networks using matlab 6.0", *Tata McGraw Hill education private limited*, (2009), pp. 187 - 188.
- [13] S. S. Panda, M. S. R. S. Prasad, MNM. Prasad, Ch. SKVR. Naidu, "Image compression using back propagation neural network", *International Journal of Engineering Science and advanced technology*, Vol. 2, No. 1, pp. 74 -78.
- [14] V. H. Gaidhane, V. Singh, Y. V. Hote, M. Kumar, "New approaches for image compression using neural network", *Journal of Intelligent Learning Systems and Applications*, Vol. 3, (2011) pp. 220 – 229.
- [15] X. Liu, H. Gu, "Hyperbolic tangent function based 2 layers structure neural network", *International conference on Electronics and Optoelectronics- IEEE*, (2011), pp. 376 – 379.
- [16] A. Bruna, "JPEG Advanced technology" available in [file:///C:/phd%20study%20material/full%20JPEG%20\(Bruna\)%20working.pdf](file:///C:/phd%20study%20material/full%20JPEG%20(Bruna)%20working.pdf) accessed on 22.09.2017
- [17] Images taken from "Computer vision test images" available at <https://www.cs.cmu.edu/~cil/v-images.html> accessed on 10.11.2017.
- [18] Standard images taken from "Image processing place" available at [http://www.imageprocessingplace.com/root\\_files\\_V3/image\\_databases](http://www.imageprocessingplace.com/root_files_V3/image_databases) accessed on 10.11.2017.