

A Novel Image Watermarking Scheme using Hybrid DWT-DCT-ELM Technique

Anurag Mishra

Department of Electronics, Deendayal Upadhyay
College, University of Delhi, New Delhi, India

Amita Goel

Department of Computer Science, Teerthanker
Mahaveer University, Moradabad, India

ABSTRACT

We propose a novel gray scale image watermarking scheme using hybrid DCT-DWT transform. The low frequency band coefficients are selected and the dataset of size 256×7 prepared using these coefficients is supplied to a newly developed simple layer feed forward network popularly known as Extreme Learning Machine (ELM). The ELM produces a normalized column vector of size 256×1 which is used as watermark sequence to be embedded in the low frequency coefficients. The DWT-DCT hybrid transform is supposed to produce better results for imperceptibility criteria. Five different image processing attacks are carried out over signed images. Watermarks are extracted successfully from signed and attacked images. Time complexity calculation indicates that the watermarking scheme is completely executed within seconds which makes it suitable for real time watermarking applications..

General Terms

Information Security, Copyright Protection

Keywords

Discrete wavelet transforms (DWT), Extreme Learning Machine (ELM), Similarity Correlation, Normalized Correlation, Peak Signal Noise Ratio (PSNR), SSIM

1. INTRODUCTION

Digital Watermarking schemes are mainly used to resolve copyright issues of the original content. The watermark may be embedded perceptibly or imperceptibly as per the requirement of the application [1]. The embedded watermark may be extracted to be matched with the embedded one to resolve the copyright disputes. Different image processing attacks can be applied over the signed images either to remove or to sabotage the watermarks [2]. Therefore, the watermark must be embedded robustly and at the same time it should not be responsible for degrading the visual quality of the signed images.

Due to this reason, the problem of watermarking is perceived as that of optimization as it trades off the twin requirement of robustness and imperceptibility [3, 4].

In this paper, we propose a hybrid DWT-DCT transform based watermarking scheme which generates a 256×1 sized normalized watermark using a newly developed single layer feed forward network (SLFN) popularly known as Extreme Learning Machine (ELM). The ELM is widely known as a neural network which is very quickly trained within millisecond time interval. This is particularly suitable for real time image and video based watermarking applications [5].

After training, the ELM produces a 256×1 sized normalized watermark sequence which is used as a watermark to be embedded with the low frequency band coefficients obtained by applying DWT-DCT hybrid transform. This transform was

used by several researchers [9, 10] and is reported to produce better results as compared to singular transform such as DWT or DCT. The visual quality of the signed images is examined using PSNR and SSIM Index. Large values of these two parameters indicate that the watermark is embedded within the low frequency band coefficients of the image in an imperceptible manner and there is hardly any effect of the embedding on the visual quality of signed images. Watermarks are extracted from the signed images and similarity correlation $SIM(X, X^*)$ and normalized correlation $NC(X, X^*)$ parameters are computed. Their large values clearly indicate that extraction process is quite successful. The three signed images are also subject to five different image processing attacks. These are JPEG with Quality = 50, 60, 70, 80, 90, Gaussian Blur (Radius = 1.0 unit), Median Filter (Filter aperture = 3.0), Gaussian Noise (10%) and Scaling ($256 \rightarrow 512 \rightarrow 256$). Watermarks are successfully recovered from the attacked images. The similarity and normalized correlation values computed over attacked images indicate that recovery is successful even from attacked images. This makes the complete watermarking scheme a robust one. The remaining of the paper is organized as follows. In section 2, Extreme Learning Machine (ELM) is briefly introduced. In section 3, embedding and extraction algorithms are explained. Experimental results are presented and discussed in section 4. Finally, the conclusion is given in section 5.

2. REVIEW OF EXTREME LEARNING MACHINE (ELM) MODEL

The Extreme Learning Machine [5, 6, 7] is a Single hidden Layer Feed forward Neural Network (SLFN) architecture. Unlike traditional approaches such as Back Propagation (BP) algorithms which may face difficulties in manual tuning control parameters and local minima, the results obtained after ELM computation are extremely fast, have good accuracy and has a solution of a system of linear equations. For a given network architecture, ELM does not have any control parameters like stopping criteria, learning rate, learning epochs etc., and thus, the implementation of this network is very simple. The main concept behind this algorithm is that the input weights (linking the input layer to the hidden layer) and the hidden layer biases are randomly chosen based on some continuous probability distribution function such as uniform probability distribution in our simulation model and the output weights (linking the hidden layer to the output layer) are then analytically calculated using a simple generalized inverse method known as Moore – Penrose generalized pseudo inverse [9].

2.1 Mathematics of ELM Model

Given a series of training samples $(x_i, y_i)_{i=1,2..N}$ and \hat{N} the number of hidden neurons where $x_i = (x_{i1}, \dots, x_{in}) \in \mathbb{R}^n$ and $y = (y_{i1}, \dots, y_{im}) \in \mathbb{R}^m$, the actual outputs of the single-

hidden-layer feed forward neural network (SLFN) with activation function $g(x)$ for these N training data is mathematically modeled as

$$\sum_{k=1}^{\hat{N}} \beta_k g(\langle w_k, x_i \rangle + b_k) = o_i, \forall i = 1, \dots, N \quad (1)$$

where $w_k = (w_{k1}, \dots, w_{kn})$ is a weight vector connecting the k^{th} hidden neuron, $\beta_k = (\beta_{k1}, \dots, \beta_{km})$ is the weight vector connecting the k^{th} hidden neuron and output neurons and b_k is the threshold bias of the k^{th} hidden neuron. The weight vectors w_k are randomly chosen. The term $\langle w_k, x_i \rangle$ denotes the inner product of the vectors w_k and x_i and g is the activation function. The above N equations can be written as

$$H\beta = O \quad (2)$$

and in practical applications \hat{N} is usually much less than the number N of training samples and $H\beta \neq Y$, where

$$H = \begin{bmatrix} g(\langle w_1, x_1 \rangle + b_1) & \dots & g(\langle w_{\hat{N}}, x_1 \rangle + b_{\hat{N}}) \\ \vdots & \dots & \vdots \\ g(\langle w_1, x_N \rangle + b_1) & \dots & g(\langle w_{\hat{N}}, x_N \rangle + b_{\hat{N}}) \end{bmatrix}_{N \times \hat{N}}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\hat{N}} \end{bmatrix}_{\hat{N} \times m}, \quad O = \begin{bmatrix} O_1 \\ \vdots \\ O_N \end{bmatrix}_{N \times m} \quad \text{and} \quad Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_N \end{bmatrix}_{N \times m} \quad (3)$$

The matrix H is called the hidden layer output matrix. For fixed input weights, $w_k = (w_{k1}, \dots, w_{kn})$ and hidden layer biases b_k , we get the least-squares solution $\hat{\beta}$ of the linear system of equation $H\beta = Y$ with minimum norm of output weights β , which gives a good generalization performance. The resulting $\hat{\beta}$ is given by $\hat{\beta} = H^+Y$ where matrix H^+ is the Moore-Penrose generalized inverse of matrix H [9]. The above algorithm may be summarized as follows:

2.2 The ELM Algorithm

Given a training set $S = \{(x_i, y_i) \in \mathfrak{R}^{m+n}, y_i \in \mathfrak{R}^m\}_{i=1}^N$, for activation function $g(x)$ and the number of hidden neurons \hat{N} ;

Step1: For $k = 1, \dots, \hat{N}$ randomly assign the input weight vector $w_k \in \mathfrak{R}^n$ and bias $b_k \in \mathfrak{R}$.

Step2: Determine the hidden layer output matrix H .

Step3: Calculate H^+ .

Step4: Calculate the output weights matrix $\hat{\beta}$ by using $\hat{\beta} = H^+T$.

Many activation functions can be used for ELM computation. In the present case, Sigmoid activation function is used to train the ELM.

2.3 Computing Moore-Penrose Generalized Inverse of a matrix

Definition 1.1:

A matrix G of order $N \times N$ is the Moore-Penrose generalized inverse of real matrix A of order if $N \times N$ $AGA = A$, $GAG = G$ and AG, GA are symmetric matrices.

Several methods, for example orthogonal projection, orthogonalization method, iterative methods and singular value decomposition (SVD) methods exist to calculate the Moore-Penrose generalized inverse of a real matrix. In ELM algorithm, the SVD method is used to calculate the Moore Penrose generalized inverse of H . Unlike other learning methods, ELM is very well suited for both differential and non – differential activation functions. As stated above, in the present work, computations are done using ‘‘Sigmoid’’ activation function.

3. GENERATING AND EMBEDDING OF WATERMARK

In the present experiment, three standard 8 bits/pixel gray scale images - Lena, Baboon and Cameraman of size 256×256 pixels each are taken as host images. The image is first decomposed into four non overlapping multi resolution subbands LL2, HL2, LH2 and HH2 by using second level DWT. The subband LL2 of size 64×64 is further divided into 4×4 pixels block and DCT of all 16 blocks is computed. Note that there are 256 such blocks in total in the entire host image. Zig-zag scanning of each 4×4 block is done to retrieve the 6 initial AC coefficients barring the first DC coefficient. Thus, a dataset of size 256×6 is created which consist of all the selected AC coefficients from all blocks. Mean of each row is taken to create a label for the row and is placed at the first column position. This creates a final dataset of size 256×7 and is used as the training dataset for the extreme learning machine. This machine is configured in the regression mode and sigmoid activation function is used for all practical computations. The machine produces a normalized output column vector of size 256×1 which is used as watermark sequence to be embedded in the low frequency band coefficients of the host image in transform domain. The embedding is carried out by using the formula given in Eqn. 4.

$$v'_i = v_i(1.0 + \alpha x_i) \quad (4)$$

where x_i is the output column vector obtained from ELM after training, v_i are the DCT coefficients and v'_i are the modified DCT coefficients which will constitute the signed image. The parameter α is known as embedding strength and is assumed as 0.25 for all our practical calculations. The obtained watermark sequence is embedded into each block and finally inverse DCT of each block is computed to obtain the signed image. Listing 3.1 gives watermark embedding algorithm.

Listing 3.1: Watermark Embedding Algorithm

1. Divide the host image into 4 non-overlapping sub-bands LL2, HL2, LH2 and HH2 using 2-level DWT
2. Divide the sub-band LL2 into 4×4 blocks and compute the DCT of all the blocks
3. Apply zigzag scan to DCT coefficients of each block
4. Select first 6 DCT coefficients from each block barring the DC coefficient and thus prepare a dataset of size 256×6
5. Calculate arithmetic mean of each row of the dataset and make it as a label column in the dataset
6. Supply this dataset to train the ELM and obtain normalized column vector to be used as watermark
7. Perform watermark embedding in the low frequency band DCT coefficients of the host image
8. Compute Inverse DCT (IDCT) to obtain the watermarked (signed) image

The signed images obtained as a result of watermark embedding are examined for visual quality by computing Peak Signal to Noise Ratio (PSNR) and SSIM Index. Note that both these quality metrics are full reference metrics which require both the reference and test images to be passed to compute the numerical value of the image quality. Large values of these metrics indicate good visual quality of the test image. These results are compiled in section 4 of this paper. The watermark sequence is extracted from the signed images. Similarity correlation parameter $SIM(X, X^*)$ and normalized correlation parameter $NC(X, X^*)$ are computed for the original

and recovered sequences. These parameters are given by Eqn. 5 and 6 respectively.

$$SIM(X, X *) = \frac{\sum_{i=1}^m \sum_{j=1}^n [X(i,j).X*(i,j)]}{\sum_{i=1}^m \sum_{j=1}^n \sqrt{X.X*}} \quad (5)$$

$$NC(X, X *) = \frac{\sum_{i=1}^m \sum_{j=1}^n [X(i,j).X*(i,j)]}{\sum_{i=1}^m \sum_{j=1}^n [X*(i,j)]^2} \quad (6)$$

Where X and X* are embedded and recovered watermark sequences respectively. Large values computed for these two parameters indicate successful watermark recovery from signed images. The watermark extraction algorithm is given in Listing 3.2.

Listing 3.2: Watermark Extraction Algorithm

1. Apply 2-level DWT to original and watermarked images
2. Divide the LL2 subband of original and watermarked image into 4*4 size blocks
3. Compute the DCT of all blocks of both the images
4. Subtract the computed coefficients of the original image from the signed image which are used in embedding and thus recover the watermark sequence
5. Compute SIM(X, X*) and NC(X, X*) correlation parameters for all signed images

4. RESULT AND DISCUSSION

4.1 Watermark Embedding and Extraction

Figure 1(a-c) show original host images – Lena, Baboon and Cameraman of size 256*256.

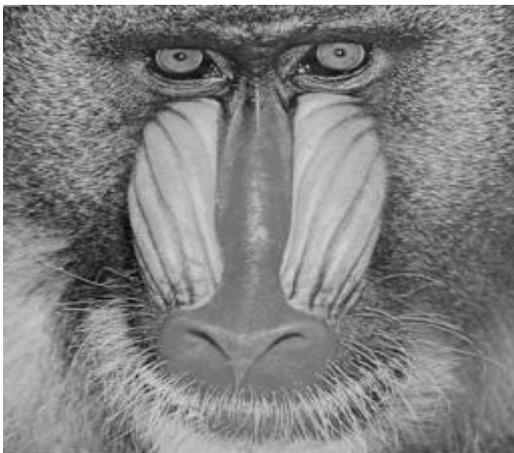


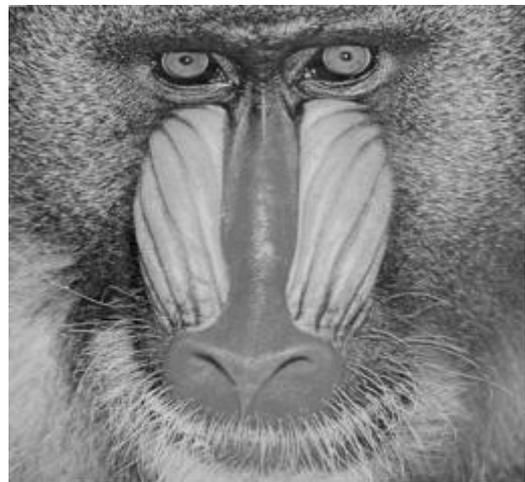
Figure 1: Original host images-(a) Lena (b) Baboon (c) Cameraman

Figure 2(a-c) show watermarked images obtained from those of Figure 1(a-c) respectively. The PSNR, SSIM, SIM and NC values obtained for these images are given on the top.

PSNR=57.4783 SSIM= 0.9998 NC=1.0000 SIM=18.9173



PSNR=59.9731 SSIM= 0.9999 NC=1.0000 SIM=20.9508



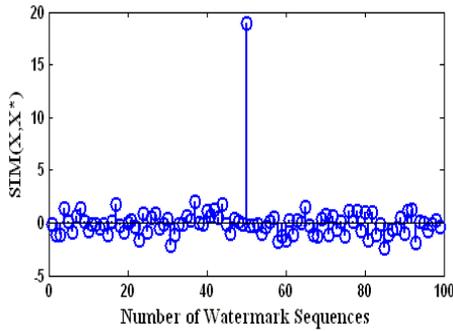
PSNR=54.6735 SSIM= 0.9993 NC=1.000 SIM=20.8687



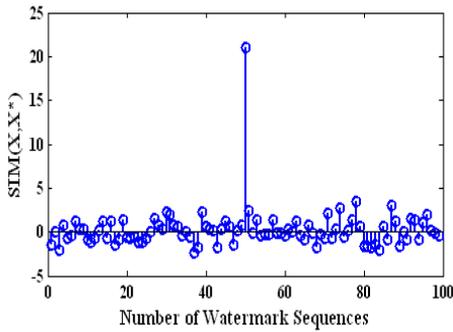
(c)

Figure 2(a-c): Signed images (a) Lena (b) Baboon and (c) Cameraman

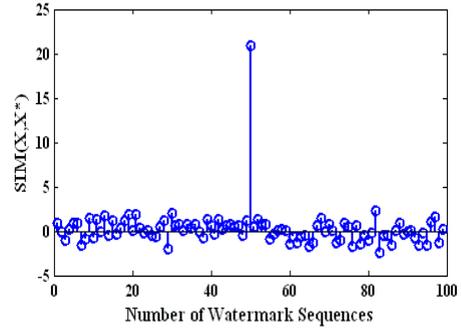
Figure 3(a-c) show the SIM plots for the watermarks extracted from the watermarked images Lena, Baboon and Cameraman respectively shown in Figure 2(a-c).



(a)



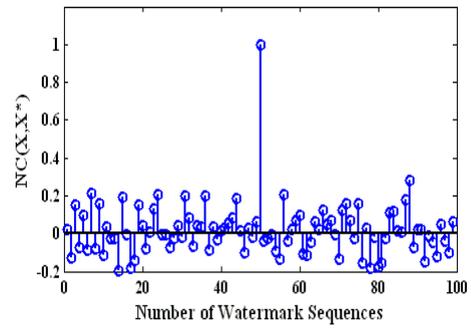
(b)



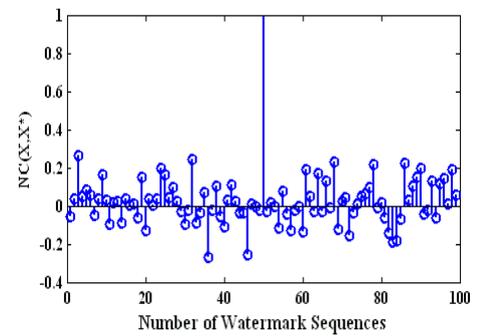
(c)

Figure 3: SIM plots for watermarks extracted from images depicted in Figure 2(a-c) respectively

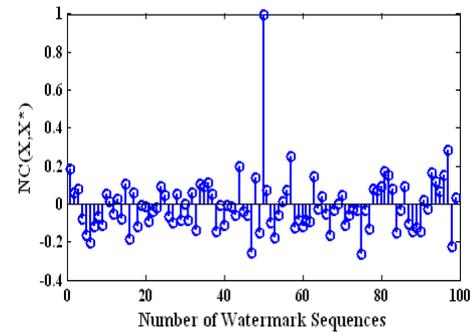
Figure 4(a-c) show the NC plots for the watermarks extracted from the watermarked images Lena, Baboon and Cameraman respectively shown in Figure 2(a-c).



(a)



(b)



(c)

Figure 4: NC plots for watermarks extracted from images depicted in Figure 2(a-c) respectively

Figure 5 shows the plot of PSNR as a function of embedding strength coefficient or scaling coefficient (α) for all three grayscale images considered for this work.

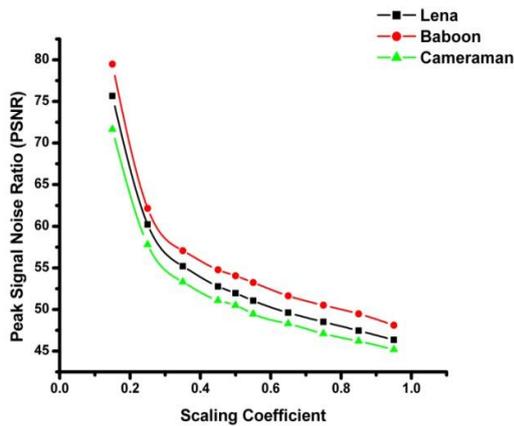


Figure 5: Plot of PSNR w. r. t Embedding Strength (α)

From the above plot, it is evident that an optimized value of α is 0.25 as on either side from it, the PSNR are found to be stabilized. In other words, if a tangent is drawn on these curves, it shall have a slope nearly equal to +1 or -1 at around $\alpha = 0.25$. This is suggestive of optimized embedding and extraction to take place at $\alpha = 0.25$. and we, therefore, consider this value of α for all our practical computations executed in the course of this experiment.

4.2 Executing Image Processing Attacks

As mentioned in section 1, the three signed images are subject to five different image processing attacks to examine the issue of robustness of the embedding algorithm. These attacks are – JPEG with quality factor (QF) = 50, 60, 70, 80 and 90, Gaussian Blur (Blur radius = 1.0), Median Filter (Filter aperture = 3.0), 10% Gaussian Noise addition and Scaling (256 -> 512 -> 256). The results of these attacks are tabulated in Table 1-3 respectively for the three grayscale images -Lena, Baboon and Cameraman.

Table 1: PSNR, SSIM, NC (X, X^*) AND SIM (X, X^*) values for attacked Lena image

Attack	PSNR (db)	SSIM	NC (X, X^*)	SIM (X, X^*)
Jpeg50	32.4790	0.9065	0.9911	8.2630
Jpeg60	33.2540	0.9160	0.9986	8.2750
Jpeg70	34.3509	0.9294	0.9917	8.2696
Jpeg80	36.0600	0.9446	0.9934	8.2673
Jpeg90	39.4833	0.9673	0.9932	8.2656
Gaussian Blur	28.8333	0.8624	0.9508	7.1993
Median Filter	30.9785	0.8934	0.9914	7.2142
Gaussian Noise	30.5642	0.7739	0.9867	7.2918
Scaling	31.5642	0.9881	0.9891	7.2670

Table 2: PSNR, SSIM, NC (X, X^*) AND SIM (X, X^*) values for attacked Baboon image

Attack	PSNR (db)	SSIM	NC (X, X^*)	SIM (X, X^*)
Jpeg50	28.9993	0.8800	0.9964	7.4407
Jpeg60	29.8439	0.8993	0.9995	7.4389
Jpeg70	31.0680	0.9215	0.9990	7.4525
Jpeg80	33.0894	0.9467	0.9994	7.4565
Jpeg90	37.5565	0.9768	0.9987	7.4594
Gaussian Blur	25.3832	0.7017	0.9456	7.3780
Median Filter	25.3105	0.7357	0.9823	7.3166
Gaussian Noise	30.5924	0.8835	0.9862	7.4636
Scaling	31.2343	0.9889	0.9801	7.4568

Table 3: PSNR, SSIM, NC (X, X^*) AND SIM (X, X^*) values for attacked Cameraman image

Attack	PSNR (db)	SSIM	NC (X, X^*)	SIM (X, X^*)
Jpeg50	39.4790	0.9122	0.9891	8.2630
Jpeg60	39.2540	0.9243	0.9986	8.2750
Jpeg70	39.3509	0.9372	0.9981	8.2696
Jpeg80	39.0600	0.9527	0.9881	7.2673
Jpeg90	39.9116	0.9726	0.9971	7.6497
Gaussian Blur	25.6943	0.8476	0.9451	7.6111
Median Filter	27.3350	0.8756	0.9956	7.6551
Gaussian Noise	30.6605	0.7313	0.9787	6.9397
Scaling	32.3080	0.9783	0.9899	7.6397

Table 4 compiles the time complexity values for all three processes carried out to develop this watermarking scheme. These are ELM training time, Watermark embedding time and Watermark extraction time. It is clear that all three time spans are within the millisecond range, however, the overall time which is the sum total of these three factors is in seconds.

Table 4: Time Spans observed for ELM training, Watermark Embedding and Watermark Extraction

Images	ELM Training Time (Sec)	Embedding Time (Sec)	Extraction Time (Sec)	Total Time
Lena	0.1563	0.5781	0.2500	0.9844
Baboon	0.1313	0.5156	0.1875	0.8344
Camera man	0.1350	0.8750	0.1563	1.1663

A careful observation of results compiled in Table 1-3 on one hand and the one given in Table 4 on the other yields following points:

1. The watermark embedding and extraction scheme carried out using DWT-DCT hybrid transform along with ELM algorithm is found to be successful as the signed images show good quality and extraction of watermarks results in large values of similarity and normalized correlation coefficients.
2. The embedding strength parameter or the scaling coefficient is optimized to equal to 0.25 which gives the optimized results for embedding and extraction.
3. The robustness studies carried out over three signed images show that their visual quality after attacks is within the expected range and watermark recovery from these images is successful. This is indicative of the fact that the watermark embedding scheme based on DWT – DCT hybrid transform along with ELM technique has been successful in producing optimized results.
4. The time complexity computations carried out in this work show that all three time spans – ELM training time, embedding time and extraction time are in the range of milliseconds. However, the overall time obtained as a sum total of all three time spans fall within seconds. This makes it explicitly clear that the proposed watermarking scheme is found to be low on time complexity which makes it suitable for real time watermarking applications – both for images and video.

5. CONCLUSIONS

In this paper, a novel grayscale image watermarking scheme is proposed which is based on hybrid DWT-DCT transform and uses a newly developed Single Layer Feed forward Network (SLFN), popularly known as Extreme Learning Machine. The DWT-DCT hybrid transform reportedly gives better results as compared to singular transforms such as DWT or DCT for different image processing applications. A dataset is developed which uses low frequency band DCT coefficients of the host images and is fed as input to the ELM. The ELM is trained in the regression mode using Sigmoid activation function and produces a normalized output column vector which is used as watermark sequence to be embedded

in the host images. The extracted watermarks show a high degree of similarity with the embedded watermark sequence. Robustness studies carried out by executing five different image processing attacks indicate that the proposed watermarking scheme is robust against selected attacks. Time complexity computations clearly indicate that all three processes – ELM training, watermark embedding and watermark extractions are finished within milliseconds. Overall time complexity is computed to be in seconds. This makes it suitable for real time watermarking applications both for images and video.

6. REFERENCES

- [1] I. J. Cox, Joe Kilian, F. Thomson Leighton and Talal Shamoan, "Secure Spread Spectrum Watermarking for Multimedia", IEEE Transactions on Image Processing, vol. 6, no.12 (1997) pp. 1673-1687
- [2] M. Kutter and F. Petitcolas, "A fair benchmark for image watermarking systems," Electronic Imaging 199: Security and Watermarking of Multimedia Content, Vol. 3657 of SPIE Proceedings, San Jose, California USA, 25-27
- [3] Charu Agarwal and Anurag Mishra, "A Novel Image Watermarking Technique using Fuzzy-BP Network", (2010) Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 102 – 105
- [4] Charu Agarwal, Anurag Mishra and Arpita Sharma, "Gray-Scale watermarking using GA-BPN Hybrid Network", (2013), Journal of Visual Communication and Image Representation, Vol 24, pp. 1135 – 1146
- [5] M-B. Lin, G-B Huang, P. Saratchandran and N. Sudararajan, "Fully complex extreme learning machine", Neurocomputing, (2005), vol (68), pp 306 – 314
- [6] G-B Huang, Q-Y Zhu and C K Siew, "Extreme Learning Machine: Theory and Applications", (2006), Neurocomputing, vol (70), pp 489-501
- [7] G-B Huang, Q-Y Zhu and C K Siew, "Real-Time Learning Capability of Neural Networks", (2006), IEEE Transactions on Neural Networks, vol 17(4), pp 863-878
- [8] G-B Huang (2004), The Matlab code for ELM is available on: <http://www.ntu.edu.sg/home/egbhuang>
- [9] S. K. Amirgholipour, A. R. Naghsh-Nilchi, "Robust Image Watermarking Based on Joint DWT-DCT," International Journal of Digital Content Technology and its Applications, vol 3, pp. 42-54, 2009
- [10] J X Liu and Z M Lu., "DWT and DCT Combined Robust Watermarking Algorithm Based on Vector Quantization with Labeled Codewords," Proc. IEEE Conf. Anti-counterfeiting, Security, and Identification in Communication (ASID'08), IEEE press, Nov. 2008, pp. 51- 54