

Novel Grayscale Image Watermarking Using Extreme Learning Machine

Anurag Mishra, Amita Goel

Abstract — A newly developed single Layer Feed Forward Network based Extreme Learning Machine (ELM) is well known for its fast learning. It has been used for different application including those in the domain of image processing. However very few researchers are using it for image watermarking based application. In this paper, a novel and robust image watermarking scheme is proposed using Extreme Learning Machine (ELM) for four grayscale images. The proposed scheme trains the ELM by using low frequency DCT Coefficients which produces a sequence of normalized 1024 real numbers used as watermark. The visual quality of the host images is evaluated using PSNR & SSIM and is found to be very good. Computed high values of Similarity Correlation and Normalized Correlation establish that the extraction process is successful. Robustness studies are carried out by applying five image processing attacks. Extracted watermarks after attacks clearly indicate that the embedding scheme is robust against the selected attacks. The complete watermarking scheme is carried out in second time span which makes it fit for good practical applications required with fast time lines.

Index Terms: Gray Scale Image Watermarking, Extreme Learning Machine (ELM), Similarity Correlation, Normalized Correlation, Real time applications

1 INTRODUCTION

THERE are certain application of multimedia which require processing with real time constraints. Video watermarking is one such application. However to standardize any watermarking scheme on a given video, it is advisable to first test it over images. Moreover, the embedding and extraction processes should be optimized to strike a balance between the twin requirements of any good watermarking Scheme-imperceptibility and robustness. Over the period of time, it has been observed that soft computing techniques such as Artificial Neural Networks(ANNs), Fuzzy Inference System(FIS), Support Vector Machines(SVMs) and evolutionary algorithms and their hybrid variants are extensively used for this purpose. However, most of these high end techniques do not give the least time complexity to be applicable to develop a good practical and real time watermarking application.

Recently, E. G. B. Huang et al. developed a fast SLFN popularly known as extreme learning machine (ELM). The benefits of this approach are that it has only one tunable parameter, namely, the number of hidden neurons, and its training process consists of only a single step, thereby reducing time up to a large extent.

- Anurag Mishra is with the Department of Electronics, Deendayal Upadhyay College, University of Delhi, New Delhi, India (e-mail: anurag_cse2003@yahoo.com)
- Amita Goel is pursuing Ph.d in Computer Science, Teerthanker Mahaveer University, Moradabad, India (e-mail: goelamita@rediffmail.com)

The training of this machine is reported to have been completed within milliseconds with a reasonably good accuracy [5].

In the present work, four standard grayscale host images - Lena, Cameraman, Goldhill and Man are used in DCT transform domain to embed a watermark sequence obtained by training the ELM. [10]

It is found that the visual quality of the signed images is good as indicated by computed PSNR values. Similarity Correlation $SIM(X, X^*)$ and Normalized Correlation $NC(X, X^*)$ parameters are calculated between embedded and extracted watermark sequences to establish successful watermark recovery.

High values of SIM and NC indicate successful recovery of the embedded watermark.

The four signed images are also subject to five different image processing attacks. These are JPEG 50,60,70,80,90, Gaussian Blur(Radius=1.0 unit), Median Filter(Filtering aperture =3.0), Gaussian Noise(10%) and Scaling(256-512-256). Watermark sequences are recovered from the attacked images. Detector responses indicate that watermark recovery is very successful from signed and attacked images as well. The paper is organized as follows. Section II of this paper gives mathematical review of Extreme Learning Machine (ELM). Section III presents the experimental details of proposed watermarking algorithm while section IV discusses the observed results and their analysis. Finally, the paper is concluded in section V.

2. REVIEW OF EXTREME LEARNING MACHINE 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 $(x_i, y_i)_{i=1,2,\dots,N}$ and \hat{N} the number of hidden neurons where $x_i = (x_{i1}, \dots, x_{in}) \in \mathfrak{R}^n$ and $y_i = (y_{i1}, \dots, y_{im}) \in \mathfrak{R}^m$, 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 modelled 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

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

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\hat{N}} \end{bmatrix}_{\hat{N} \times m}, \quad O = \begin{bmatrix} o_1 \\ \vdots \\ o_N \end{bmatrix}_{N \times m}$$

$$Y = \begin{bmatrix} y_i \\ \vdots \\ y_N \end{bmatrix}_{N \times m} \quad (3)$$

and

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

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

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 $\hat{\beta} = H^+T$.

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 the Moore-Penrose Generalized Inverse of a matrix

Definition 1.1:

A matrix G of order $\hat{N} \times N$ is the Moore-Penrose generalized inverse of real matrix A of

order if $N \times \hat{N}$ $AGA = A, GAG = G$ 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. EXPERIMENTAL DETAILS

3.1. Watermark Generation and Embedding

In this experiment, four grayscale images - Lena, Cameraman, Goldhill and Man of size 256×256 pixels each are taken as host images. The image object is divided into 8×8 pixel blocks and DCT of all such blocks is computed to transform the blocks into frequency domain. Zigzag scanning of each block is done to select first 21 AC coefficients barring the DC coefficient. Thus, a dataset of size 1024×21 is created which holds 21 selected coefficients from each of 1024 blocks in all. From this dataset, the mean of each row is computed and placed at the first column position as label. This results in creation of another dataset of size 1024×22 . This is used as the training dataset for the ELM to be used in regression mode. Sigmoid activation function is used to train the ELM. We used hidden neurons =60 to optimize the watermarking scheme. After training, an output vector of size 1024×1 is obtained from the ELM. This output vector is used as watermark to be embedded within the host image using Eqn. 4.[10]

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

where x_i is output obtained from ELM after training, v_i are DCT coefficients and v'_i are coefficients of the signed image. The parameter α is known as embedding strength and is optimized to be 0.3 for all our practical calculations. The computed watermark is embedded into each block and

the inverse DCT of each block is taken to the signed image. Listing 1 shows watermark embedding algorithm.

Listing 1: Watermark Embedding Algorithm

1. Block wise transform given grayscale host image by using DCT algorithm
3. Apply zigzag scanning to all AC coefficients of each block and select first 21 coefficients from each block barring the DC coefficient and develop a dataset of size 1024×21 using these coefficients
4. Compute the mean of all 21 coefficients for each row and place it in first column as label. Thus, recreate a dataset of size 1024×22
5. Train the ELM in regression mode by supplying this dataset to the machine. As a result, the ELM produces an output vector of size 1024×1 which is used as watermark to be embedded within the host image using Eqn. 4.
6. Take Inverse DCT (IDCT) to obtain signed image

The block diagram of the embedding process is shown in Fig. 1.

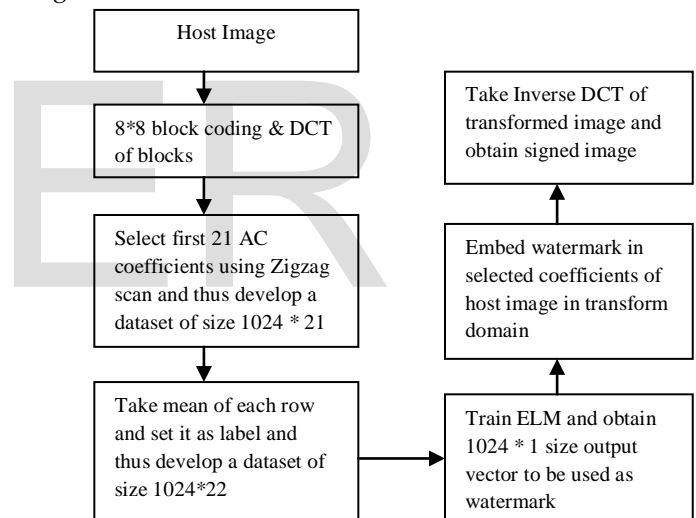


Fig. 1: Watermark Embedding Procedure

3.2 Extracting the Watermark:

In the extraction process, first of all, 8×8 block wise DCT of both host and signed images are computed and the coefficients of the original image which are used in embedding process are subtracted from the respective coefficients of the signed image. In this manner, both the original and recovered watermark sequences X and X^* are known. A statistical similarity correlation and normalized correlation check is performed over X and X^* as given by Eqn. 5 and Eqn. 6. Listing 2 depicts the watermark extraction algorithm.

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

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

Listing 2: Watermark Extraction Algorithm

1. Divide both the original and watermarked images into 8*8 size blocks
2. Compute DCT of all blocks of the images
3. Subtract only those computed coefficients of the original image from the respective coefficients of signed image which are used in embedding process and thus recover the watermark X*
4. Compute the SIM(X, X*) correlation parameter

The block diagram for extraction process is shown in Fig. 2.

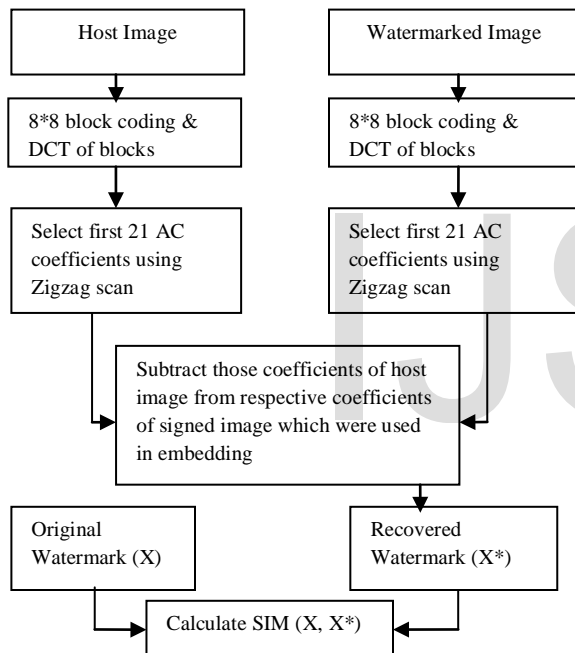


Fig. 2: Watermark Extraction Procedure

4. RESULTS AND DISCUSSION

4.1 Embedding and Extraction

Figure 3(a-d) show gray scale host images Lena, Cameraman, Goldhill and Man of size 256x256. As indicated in section III(A), the output column vector of ELM is first normalized to N(0, 1) and subsequently embedded within these images to obtain signed images depicted in Fig. 4(a-d) respectively using Eqn. 4. The visual quality of signed images is ascertained by computing PSNR, SSIM parameters. The extracted watermark sequences (X*) are matched with embedded one(X). Similarity Correlation SIM(X, X*) and Normalized Correlation NC(X, X*) coefficients are computed for this

purpose. The respective PSNR, SSIM, NC and SIM values are mentioned above these images.

High computed PSNR values indicate that the visual quality of these images is very good. Fig. 5(a-d) respectively shows the SIM plots for the watermarks recovered from signed images of Fig. 4(a-d).



(a)



(b)



(c)

PSNR=63.1654 SSIM=0.9999 NC=1.0000 SIM=14.7195



(b)



(d)

PSNR=66.0262 SSIM=1.0000 NC=1.0000 SIM=18.1022



(c)

Fig 3: Original host Images – (a) Lena (b) Cameraman
(c) Goldhill (d) Man

PSNR=64.2327 SSIM=0.9999 NC=1.0000 SIM=16.8512

PSNR=63.7381 SSIM=0.9999 NC=1.0000 SIM=16.9206



(a)



(d)

Fig 4: Signed Images – (a) Lena and (b) Cameraman

(c) Goldhill (d) Man

Fig 5(a-d) respectively show $SIM(X, X^*)$ plots/detector responses for four signed images. Similarly $NC(X, X^*)$ values are also computed for these signed images. A close observation of $SIM(X, X^*)$ and $NC(X, X^*)$ parameters indicates a high degree of similarity between embedded and recovered watermarks, thereby indicating a successful extraction process used in the proposed scheme. Note that the embedding is carried out using eqn (4). α is known as embedding strength or scaling coefficient. Although Cox et al [10] have stressed upon the use of multiple scaling factors due to variable nature of object statistics available within the host image, we however use simple scaling factor (α) to carry out optimized embedding thereby balancing out visual quality and robustness. The optimization of the watermark embedding strength α is depicted in Fig 6.

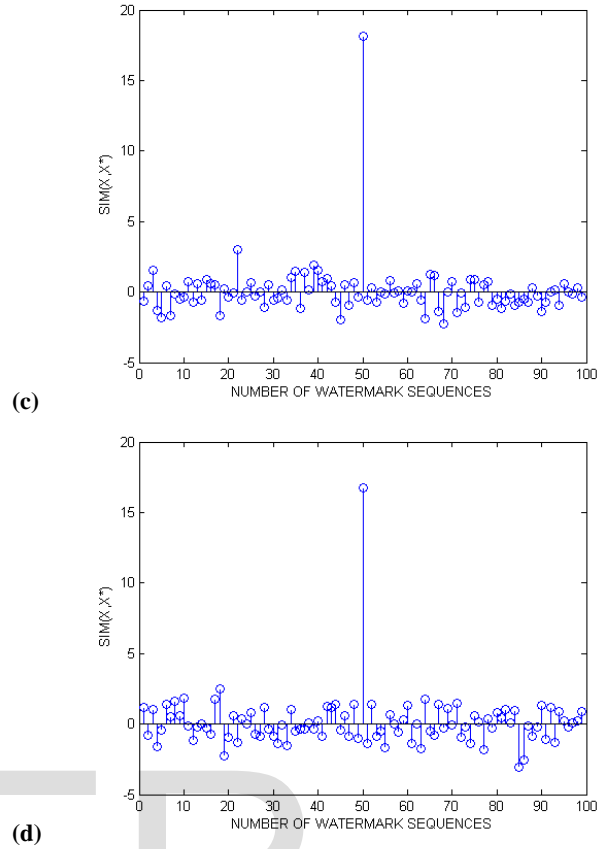
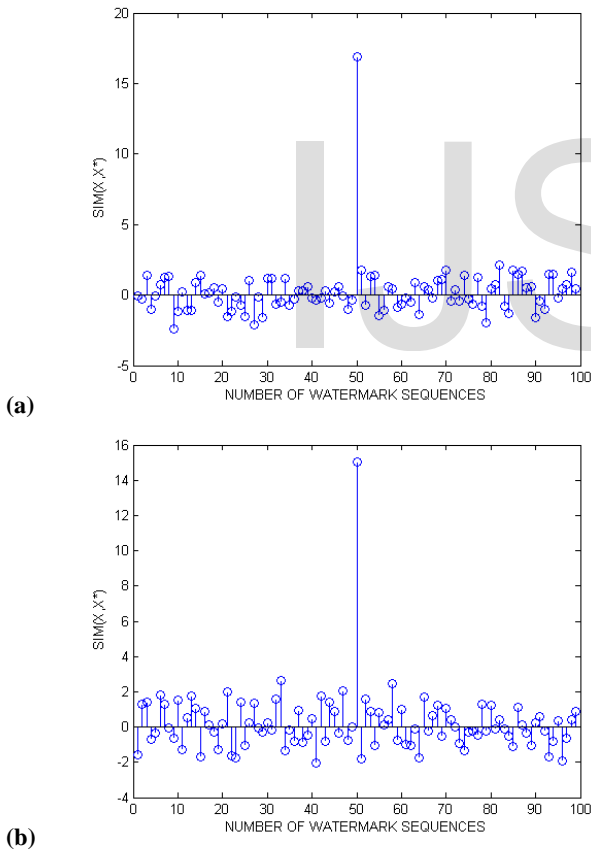


Fig. 5: SIM plots for watermarks extracted from images depicted in Fig. 4(a), (b), (c) and (d) respectively

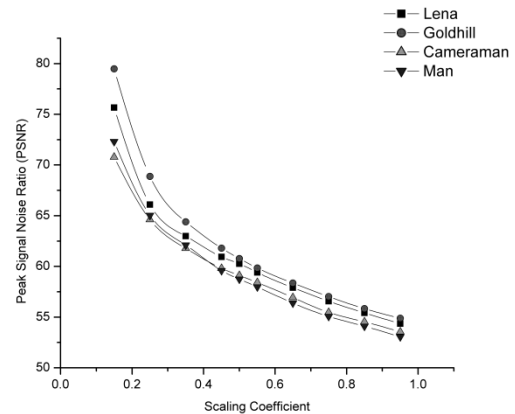


Fig. 6: Plot of PSNR with respect to Scaling Coefficient (α)

From the above plot, it is evident that an optimized value of α is 0.3 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 $\alpha = 0.3$. This is suggestive of optimized embedding and extraction to take place at $\alpha = 0.3$ and we, therefore,

consider this value of α for all our practical computations executed in the course of this experiment.

This work may further be extended to watermarking of video sequences as it compulsorily requires embedding and extraction of watermarks within the given timelines

4.2 Executing Image Processing Attacks:

The robustness studies have been carried out over signed images by executing five different image processing attacks JPEG 50,60,70,80,90,Gaussian Blur(Radius=1.0 unit), Median Filter(Filter aperture=1.0), Gaussian Noise (10%),Scaling(256-512-256).

Watermark sequences are extracted from these attacked images and PSNR, SSIM, NC(X,X*), SIM(X,X*) values are computed and compiled in table 1.

TABLE 1

SIM(X, X *) AND NC(X, X *) VALUES FOR ATTACKED IMAGES

Images	Attack	PSNR(db)	SSIM	NC (X,X*)	SIM (X,X*)
Lena	Jpeg50	32.4287	0.9146	0.9973	11.6780
	Jpeg60	33.2533	0.9263	0.9992	11.6658
	Jpeg70	34.3508	0.9408	1.0002	11.7017
	Jpeg80	36.0619	0.9580	0.9996	11.6811
	Jpeg90	39.4891	0.9791	0.9994	11.6829
	Gaussian Blur	29.3929	0.8566	0.8714	11.3424
	Median Filter	30.9789	0.8933	0.9464	11.3348
	Gaussian Noise	21.4847	0.5897	0.9826	11.5820
	Scaling	38.2007	0.9832	0.9868	11.6828
	Camera man	Jpeg50	31.7438	0.9121	0.9971
Jpeg60		32.6296	0.9243	0.9981	10.7524
Jpeg70		33.8537	0.9372	0.9991	10.7571
Jpeg80		35.8003	0.9527	0.9995	10.7621
Jpeg90		39.9089	0.9726	0.9999	10.7626
Gaussian Blur		26.2465	0.8642	0.8338	10.1659
Median		27.3342	0.8756	0.9323	10.2706

	Filter				
	Gaussian Noise	21.7100	0.4584	0.9729	10.7503
	Resize	36.6365	0.9837	0.9876	10.6857
Goldhill	Jpeg50	30.5968	0.8686	0.9966	13.8263
	Jpeg60	31.2197	0.8843	0.9967	13.8579
	Jpeg70	32.1664	0.9042	0.9968	13.8596
	Jpeg80	33.7967	0.9300	0.9975	13.8687
	Jpeg90	37.6095	0.9669	0.9971	13.8572
	Gaussian Blur	27.9950	0.7853	0.8193	12.9931
	Median Filter	28.0080	0.7847	0.8911	12.8636
	Gaussian Noise	21.5312	0.5461	0.9821	13.7444
	Resize	36.5114	0.9754	0.9768	13.6164
Man	Jpeg50	31.5059	0.9059	0.9930	12.6453
	Jpeg60	32.3231	0.9160	0.9943	12.6235
	Jpeg70	33.5032	0.9294	0.9955	12.6113
	Jpeg80	35.2745	0.9446	0.9966	12.6119
	Jpeg90	39.0240	0.9673	0.9971	12.6223
	Gaussian Blur	27.6520	0.8773	0.8586	12.0754
	Median Filter	29.6085	0.8934	0.9321	12.1560
	Gaussian Noise	21.9249	0.4636	0.9767	12.5814
	Resize	35.9352	0.9853	0.9799	12.5668

TABLE 2:

TIME SPANS FOR ELM BASED WATERMARKING

Images	ELM Training Time (Sec)	Embedding Time (Sec)	Extraction Time (Sec)	Total Time
Lena	0.0411	1.5000	0.0546	1.5957
Cameraman	0.0519	1.5020	0.0531	1.6070
Goldhill	0.0525	1.6702	0.0378	1.7605

Man	0.0681	1.7100	0.0694	1.8604
-----	--------	--------	--------	--------

A careful observation of the values compiled in table 1 yields following points.

1. The recovery of watermark sequences is successful from all four images. This is attributed to high computation values of $SIM(X,X^*)$ and $NC(X,X^*)$. However, the recovery is the best in case of GoldHill, followed by Man, Lena and Cameraman.
2. PSNR and SSIM values computed to examine the imperceptibility of attacked images are all high and above required thresholds. This clearly indicates that visual quality after attacks is good
3. As both the visual quality of attacked images and computed values of SIM/NC are high. The proposed watermarking scheme is found to be the optimized one: This optimization is attributed to the chosen value of the embedding strength α parameter which is selected to be $\alpha = .3$.
4. The embedding and extraction time spans of the proposed scheme is in the range of seconds. This makes it quite suitable for further developing real time applications of image watermarking.

5. CONCLUSION

A novel image watermarking algorithm based on training of a fast neural network, known as Extreme Learning Machine (ELM) is proposed in this paper. To the best of our knowledge, we have used this machine for image watermarking in regression mode for the first time. Two grayscale images – Lena and Baboon are embedded with the output of the trained ELM within the selected low frequency coefficients obtained after 8×8 block coding followed by DCT of the blocks. Visual quality of signed images is examined by PSNR. High PSNR values of signed images clearly indicate that embedding

process is well optimized and the visual quality after embedding is quite good. Watermark extraction is performed using Cox's algorithm and SIM plots between the original and recovered watermarks are also obtained. High SIM values indicate that watermark extraction is also successful. Overall, the ELM training, embedding and extraction processes are well optimized and the algorithm finds good practical applications, especially in situations that require fulfilling time constraints such as video watermarking.

6. REFERENCES

- [1] 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
- [2] Rajesh Mehta, Anurag Mishra, Rampal Singh, Navin Rajpal, "Digital Image Watermarking in DCT Domain Using Finite Newton Support Vector Regression" (2010) Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp 123 - 126
- [3] Mukesh C. Motwani and Fredrick C Harris, Jr, "Fuzzy Perceptual Watermarking for Ownership Verification" (2009) Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV09), Las Vegas, Nevada, July 13-16
- [4] Der-Chyuan Lou, Ming-Chiang Hu, and Jiang-Lung Liu, "Healthcare Image Watermarking Scheme Based on Human Visual Model and Back-Propagation Network", (2008) Journal of C.C.I.T, vol.37, no.1, pp. 151-162
- [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] D. Serre (2002), "Matrices: Theory and Applications", Springer Verlag, New York Inc
- [10] Ingemar J. Cox, Joe Kilian, F. Thomson Leighton and Talal Shamoan, "Secure spread spectrum watermarking for multimedia", IEEE Transactions on Image Processing, vol 6(12), (1997), pp 1673-168