ISSN No. 0976-5697
Volume 8, No. 3, March – April 2017
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

A Novel Blind Digital Watermarking Scheme Based on SVD and Online Sequential Extreme Learing Machine

Neelam Dabas
Department of Computer Science
University of Delhi
Delhi, India

Geetanjali Kher
Department of Computer Science
KMC, University of Delhi
Delhi, India

RamPal Singh
Department of Computer Science
DDUC, University of Delhi
Delhi, India

Vikash Chaudhary
Department of Computer Science
BNC, University of Delhi
Delhi, India

Abstract: **With the development of internet, it is very easy to distribute and access the digital media. But with the distribution, security of intellectual property becomes a critical issue. Watermarking is one of the primary tool to protect intellectual property. A novel digital watermarking scheme has been proposed by using SVD and OSELM in IWT domain. This proposed scheme is divided into three phases. During the first phase of the embedding process, the cover image is transformed through Integer wavelet transformation (IWT) to get the low and high energy coefficient. The LL sub-band is divided into non-overlapping coefficients blocks and applied to SVD to get the singular values. Then watermark is embedded to modify these singular values. Second phase is the training phase in which an Online Sequential Extreme Learning Machine(OSELM) is trained to learn the relationship between the original coefficient and their watermarked version. The third and the final phase is the extraction phase in which watermark is extracted from the image by using the trained OSELM. This proposed method is the blind method as no original cover image is required in the extraction phase. Experiments have been performed on various images by attacking them with noise, cropping, blurring etc. Peak Signal to Noise Ratio(PSNR) and Bit Error Rate(BER) are calculated. Less BER value show that extracted watermark logo is very much similar to the original one and works efficiently to prove ownership.**

Keywords: **IWT, SVD, ELM, OSELM, PSNR, BER**

I. INTRODUCTION

Digital watermark is the tag or label which conveys the information about the owner for copyright protection of the image or any other information having some utility[1]. It is a layer of security for digital content i.e. image, audio, video etc. Applications of watermarking includes ownership assertion to verify the identity of the owner, tampering detection to protect the integrity of the media, copyright control for commercial distribution, broadcast monitoring to make sure a particular media is transmitted on a broadcast channel and transaction tracking to track the misuse of information[1]. The primary requirement of any digital watermarking technique is imperceptibility and robustness where imperceptibility requires that the embedding of watermark should not distort the quality of original image. Robustness requires that the image processing should not alter or destroy the watermark. There are two types of watermarking schemes blind and non-blind digital watermarking. In case of non-blind watermarking schemes, during their extraction phase, there is the need to have the original cover image but in case of blind digital watermarking schemes, watermark can be extracted without the presence of original cover image. Our proposed algorithm is considered as the blind digital watermarking scheme as no original cover image is required during the extraction process.

SVD is a popular technique used in watermarking. Earlier Liu & Tan [2] proposed a blind digital watermarking scheme based on SVD. But this method was not able to resist the various rotation, scaling and print-scan attacks. Ghazy et al. [3] embedded the watermark into the singular values of non-overlapping blocks. This method is performed efficiently under various attacks but failed to perform with geometric attacks. After these SVD is used with various frequency domains like DCT, DWT etc. SVD with DCT had been proposed by Quan&Qingsong[4]. A similar method has been proposed by Ganic&AhmetEskicioglu[5] in which SVD is combined with SVD. All SVD based watermarking schemes worked efficiently.

Our proposed algorithm is based on SVD and Online Sequential Extreme Learning Machine (OSELM) in Integer Wavelet Transformation (IWT). OSELM is one of the variant of ELM which is based on Single Layer Feedforwad Neural Network (SLFN) [6]. Rest of the paper is organized as: Explanation of background theories of each key terms used in this algorithm is described in Section two. Proposed algorithm is explained in section three. All the experimental results are given in Section four followed by the Conclusion section.

II. A BRIEF LITERATURE SURVEY OF KEY TERMS

A. Integer Wavelet Transformation

In Integer wavelet transformation (IWT), an integer data set is converted into another integer data set [7]. As in DWT, after IWT transformation, image is divided into 4 sub-bands *LL, HL, LH, HH*. One disadvantage of DWT transformation is that the perfect reconstruction of image is not possible because the wavelet coefficients have the floating point value. Because of the floating point precision problem, perfect reconstruction of image is not possible[8]. IWT overcomes this problem as all the coefficient set are of integer type and during the reverse transformation, there will be no loss of information [9][10]

Lifting scheme algorithm is used to perform the IWT. This algorithm is divided into three steps: Split, Predict and Update [9]

1. Split: Divide the samples into even and odd samples.
2. Predict: Predict the odd samples from the even one.
3. Update: Update the data in even set with the data from the predicted odd set.

B. Singular Value Decomposition

Singular Value Decomposition, a linear algebra tool to decompose or factorize a matrix, rectangular or square, into three sub-matrices. Given a*m* x *n* matrix *A*. This matrix is decomposed [11][12][13]into three sub-matrices known as singular value decomposition of *A*. Matrix *A* is transformed as

$$A_{m\times n} = [U_{m\times m}\emptyset_{m\times n}V_{n\times n}] \quad (1)$$

$$= [u_1, u_2, \ldots, u_m] \times \begin{pmatrix} \emptyset_1 & 0 & 0 \\ 0 & \emptyset_2 & 0 \\ 0 & 0 & \emptyset_n \end{pmatrix} \times [v_1, v_2, \ldots, v_n] \quad (2)$$

Where Φ is a diagonal matrices (the matrices whose diagonal entries are non-zero and all other entries are zero). All diagonal entries are non-negative and arranged in decreasing order such that

$$\emptyset_1 > \emptyset_2 > \cdots > \emptyset_n \quad (3)$$

Diagonal entries are positive square root of eigen values of AA^T. These entries are known as singular values of matrix *A*. *U* of size *m* x *m* and *V* of size *n* x *n* are known as orthogonal matrix (the matrix whose transpose is equal to its inverse). The column of orthogonal matrix *U* are the left singular vectors, which are the eigen vectors of AA^T. The columns of orthogonal matrix *V* are the right singular vectors which are the eigen vectors of A^TA. Each singular value specifies the luminance of an image while the singular vectors specifies the geometry of the image[13]. SVD based image processing techniques were focused in compression, watermarking and quality measures [5][13][14][15][16] since by using some singular values, a image with very small difference from the original one can be regenerated. SVD is popular for the watermarking [13][17] because

1. Few singular values can represent large portion of signal energy.
2. SVD can be applied to square and rectangular images
3. The SV's (singular values) of an image have very good noise immunity, i.e., SV's do not change significantly when a small perturbation is added to an image intensity values
4. SV's represent intrinsic algebraic properties.

C. Extreme Learning Machine

ELM is a single layer feed forward network developed by Huang et al.[6].ELM is considered as the batch learning algorithm in which all training data samples are fed once to the learning algorithm. Consider a set of $\mathbb{Z}$ samples $(x_i, t_i) \in R^N \times R^M$ where x_i is $N \times 1$ input data and t_i is M$\times 1$ target data .The output of SLFN with *L* hidden nodes can be represented by

$$f_{ELM}(x) = \sum_{i=1}^{L} \xi_i g_i(x) = \sum_{i=1}^{L} \xi_i G(w_i, b_i, x) = o_i \quad (4)$$

Where w_i and b_i are randomly generated learning parameters of hidden node. ξ_i is the weight connecting the i^{th} hidden node to the output nodes, b_i is the bias. $g(x)$ is the activation function. For additive hidden nodes $g(x)$ is defined as

$$G(w_i, b_i, x) = g(w_i \cdot x + b_i) \quad (5)$$

Where $w_i \cdot x$ denotes the inner product of vector w_i and *x* in R^N. Equaion (4) can be re-written as

$$\mathbb{M}\xi = T \quad (6)$$

Where $\mathbb{M}$ is the hidden layer output matrix, ξ is the weight vector and T is the target data. ξ is estimated as

$$\xi = \mathbb{M}^{\dagger}T \quad (7)$$

Where $\mathbb{M}^{\dagger}$ is the Moore-Penrose generalized pseudo inverse [18]

Essence of ELM

The basic essence of ELM is that:

1. There is a random selection of learning parameters ξ_i andb_i[19][20]. No iterative tuning is required as done in back propagation algorithm.
2. $||\mathbb{M}\xi - T||$, the training error and $||\xi||$, norm of output weight need to be minimized[19][20][21]
3. Least square method is used to calculate ξ_i between the hidden layer and the output layer.

Algorithm

ELM algorithm is as follows: For $\mathbb{Z}$ training samples$(x_i, t_i) \in R^N \times R^M$, *L* number of hidden neurons and *g(x)* as an activation function

1) Input weight w_i and bias b_i are randomly generated, where $i = 1, \ldots, N$
2) Calculate $\mathbb{M}$, hidden neuron output matrix.
3) Calculate ξ, output weight using equation $\xi = \mathbb{M}^{\dagger}T$ as defined in equation (8)

ELM is considered as a batch learning algorithm in which all the input samples are fed to the algorithm once. And if new data comes then there is a need to retrain all data set again.

D. Online Sequential Extreme Learning Machine

As discussed in[22], the said algorithm is known as online sequential extreme learning machine (OSELM) in which data may be learned one-by-one or trunk-by trunk. Block of data is not used again which has already been used for training of machine model. Hence, sequential learning algorithm may be summarized as [22]:

1. Training data is sequential in one-by-one or block-by-block. Block size may be fixed or can vary in size.
2. Only the newly arrived data samples are used for training, no previous data is used for training.
3. Training data sample is discarded once it is used for training of machine.
4. Learning machine has no advance knowledge that how much training data sample to be presented to learning algorithm.

OSELM consists of two phases: Initialization phase and sequential learning phase. A initial hidden layer output matrix

$\mathbb{M}_0$ is generated in initialization phase. This matrix is used in sequential phase to implement OSELM. $rank(\mathbb{M}_0) = L$, the number of hidden neurons. So distinct training data are required in initialization phase followed by sequential learning phase where data can come one-by-one or chunk-by-chunk. All training data are used only once. No training data is used again.

A) Initialization Phase:

Consider the training samples $\mathbb{Z} = (x_i, t_i) \in R^N \times R^M$

1. Take the initial training data $\mathbb{Z}_0$ from the given training samples $\mathbb{Z}_0 = (x_i, t_i)_{i=1}^{z_0}$ where $z_0 \geq L$ (the number of hidden nodes)
2. For additive hidden nodes, randomly generate input weight w_i and bias b_i. For RBF hidden nodes, generate the center w_i and impact factor $b_i, i = 1, \cdots, L$
3. Calculate the initial hidden layer output matrix $\mathbb{M}_0$

$$\mathbb{M}_0 = \begin{bmatrix} G(w_1, b_1, x_1), & \cdots & G(w_L, b_L, x_1) \\ \vdots & \vdots & \vdots \\ G(w_1, b_1, x_{z_0}), & \cdots & G(w_L, b_L, x_{z_0}) \end{bmatrix}_{Z_1 \times L}$$

4. Calculate the initial output weight. According to Equation (7) of ELM, ξ is given by $\xi = \mathbb{M}^{\dagger} T$

$$\mathbb{M}^{\dagger} = (\mathbb{M}^T \mathbb{M})^{-1} \mathbb{M}^T \qquad (8)$$

Substitute (8) into (7)

$$\xi = (\mathbb{M}^T \mathbb{M})^{-1} \mathbb{M}^T T \qquad (9)$$

The sequential implementation of (9) give the result of OSELM. As in [22], the initial learning output weight is $\xi^{(0)} = P_0^{-1} \mathbb{M}_0^T T_0$ where $P_0 = \mathbb{M}_0^T \mathbb{M}_0$

B) Sequential Learning Phase:

1. Take another z_1 number of observations in the next chunk of data.

$$z_1 = (x_i, t_i)_{i=z_0+1}^{z_0+z_1} \qquad (10)$$

Next learning output weight is given by

$$\xi^{(1)} = P_1^{-1} \begin{pmatrix} \mathbb{M}_0 \\ \mathbb{M}_1 \end{pmatrix}^T \begin{pmatrix} T_0 \\ T_1 \end{pmatrix} \qquad (11)$$

Where

$$P_1 = \begin{pmatrix} \mathbb{M}_0 \\ \mathbb{M}_1 \end{pmatrix}^T \begin{pmatrix} \mathbb{M}_0 \\ \mathbb{M}_1 \end{pmatrix} = (\mathbb{M}_0^T \mathbb{M}_1^T) \begin{pmatrix} \mathbb{M}_0 \\ \mathbb{M}_1 \end{pmatrix}$$

$$= P_0 + \mathbb{M}_1^T \mathbb{M}_1 \qquad (12)$$

As in (23), equation (11) can be formulated as

$$\xi^{(1)} = \xi^{(0)} + P_1^{-1} \mathbb{M}_1^T (T_1 - \mathbb{M}_1 \xi^{(0)}) \qquad (13)$$

2. Take *(k+1)th* chunk of new data
3. For the *(k+1)th* iteration, calculate the partial hidden layer output matrix $\mathbb{M}_{k+1}$
4. Calculate

$$T_{k+1} = \begin{pmatrix} t_{(\sum_{j=0}^{k} z_j)+1} \\ \vdots \\ t_{(\sum_{j=0}^{k+1} z_j)} \end{pmatrix}^T_{Z_{k+1} \times L} \qquad (14)$$

5. Calculate the output weight by

$$\xi^{(k+1)} = \xi^{(k)} + P_{k+1}^{-1} \mathbb{M}_{k+1}^T (T_{k+1} - \mathbb{M}_{k+1} \xi^{(k)}) \qquad (15)$$

Where $P_{k+1} = P_k + \mathbb{M}_{k+1}^T \mathbb{M}_{k+1}$ (16)

By using Woodbury formula

$$P_{k+1}^{-1} = (P_k + \mathbb{M}_{k+1}^T \mathbb{M}_{k+1})^{-1}$$

$$= P_k^{-1} + P_k^{-1} \mathbb{M}_{k+1}^T (I + \mathbb{M}_{k+1} P_k^{-1} \mathbb{M}_{k+1}^T)^{-1} \times \mathbb{M}_{k+1}^T P_k^{-1}$$

Let us substitute $P_{k+1}^{-1} = C_{k+1}$ in the above equation, then

$$C_{k+1} = C_k - C_k \mathbb{M}_{k+1}^T (I + \mathbb{M}_{k+1} C_k \mathbb{M}_{k+1}^T)^{-1} \times \mathbb{M}_{k+1}^T C_k$$

$$\xi^{(k+1)} = \xi^{(k)} + C_{k+1} \mathbb{M}_{k+1}^T (T_{k+1} - \mathbb{M}_{k+1} \xi^{(k)}) \qquad (19)$$

Remarks from [22] are given by :

1. The chunk size of each training samples need not to be fixed, it may vary from one iteration to another i.e. the training size of $\mathbb{Z}_{i+1}$ samples need not to be same as $\mathbb{Z}_i$. If chunk size is 1, then according to Sherman-Morrison formula

$$K_{k+1} = K_k - \frac{K_k h_{k+1} h_{k+1}^T K_k}{1 + h_{k+1}^T K_k h_{k+1}} \qquad (20)$$

$$\xi^{(k+1)} = \xi^{(k)} + K_{k+1} h_{k+1} (t_{k+1}^T - h_{k+1}^T \xi^{(k)}) \qquad (21)$$

Where

$$h_{k+1} = G(w_1, b_1, x_{(k+1)}), \cdots, G(w_L, b_L, x_{(k+1)})$$

2. The special case of OSELM where all the training data samples are present in only one iteration. So if the initial training data samples number is equal to N the total no. of data samples $\mathbb{Z}_0 = \mathbb{Z}$ then OSELM transforms into batch learning ELM

III. WATERMARKING SCHEME

This algorithm consists of three parts: Watermark embedding, ELM Training and Watermark extraction

A. *Watermark Embedding*

The block diagram of watermark embedding is shown in Figure.

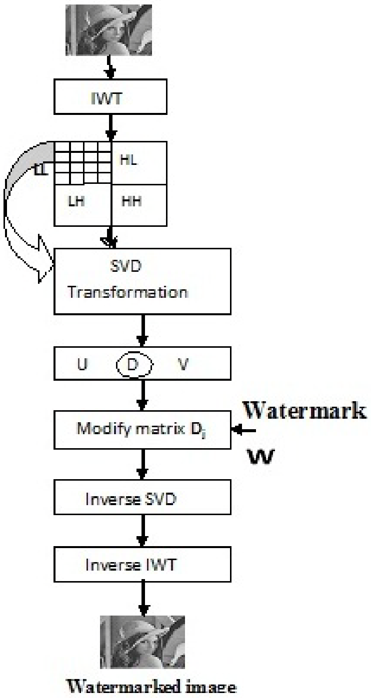


Figure 1. Proposed Watermark Embedding Process.

Let us assume there is a cover image *C* of size $N \times N$ and a binary watermark image *W* with size $P \times P$. As it is a binary image so $W_{ij} = \{0,1\}$. The embedding algorithm is:

1. 1-level IWT is applied to the cover image C to get four sub-bands denoted as *LL*, *HL*, *HL*, *HH*sub-bands.
2. Take *LL*sub-band and divide it into 4×4 non overlapping coefficient blocks $LL_{ij}, i \leq N, j \leq N$
3. SVD is applied to each 4×4 block. We will get three sub-matrices of each block
$$LL_{ij} = U_{ij} \emptyset_{ij} V_{ij}^T \quad (22)$$
4. Take out the four singular values of each LL_{ij}, $\emptyset_{ij} = (\emptyset_1, \emptyset_2, \emptyset_3, \emptyset_4)$
5. To get the modified $\emptyset'_{ij}$, modify the value of $\emptyset_2$ based on the basis of following mathematical operations, where α is the embedding strength of watermarking scheme.
$$\emptyset_2 = \begin{cases} \mu + \alpha \times \delta, & if\ W_{ij} = 1\ and\ \emptyset_2 < \mu \\ \emptyset_2, & if\ W_{ij} = 1\ and\ \emptyset_2 > \mu \\ \mu - \alpha \times \delta, & if\ W_{ij} = 0\ and\ \emptyset_2 > \mu \\ \emptyset_2, & if\ W_{ij} = 0\ and\ \emptyset_2 < \mu \end{cases}$$
$$\mu = \frac{1}{4} \times \sum_{i=1}^{4} \emptyset_i$$
$$\delta = \frac{1}{4} \times \sum_{i=1}^{4} (\emptyset_i - \mu)^2 \quad (23)$$
6. Perform inverse SVD to get the modified coefficient block
$$U_{ij} \emptyset'_{ij} V_{ij}^T = LL'_{ij} \quad (24)$$
7. Perform inverse *IWT* with LL'_{ij}, HL, LH, HH to get the final watermarked image *C'*

B. *OSELM Training*

The block diagram to train an OSELMis shown in figure 2.

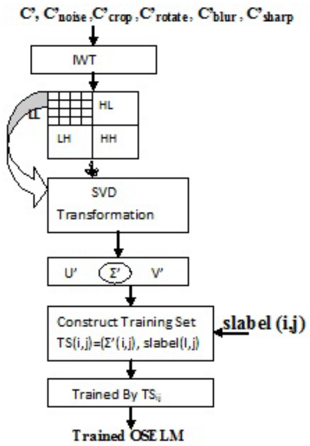


Figure 2. OS-ELM Training Process.

Training set is derived from two sources, first is the original watermarked image *C'* and second one is the corrupted watermarked image by Gaussian noise C'_{noise}, cropping C'_{crop}, rotation C'_{rotate}, blurring C'_{blur}, sharpening C'_{sharp} respectively. Training of OS-ELM is as follows:

1. 1-level IWT is performed on each image, original watermarked and all the corrupted images $C', C'_{noise} C'_{crop} C'_{rotate} C'_{blur} C'_{sharp}$ and get their respective *LL* sub-bands as $LL', LL'_{noise} LL'_{crop} LL'_{rotate} LL'_{blur} LL'_{sharp}$
2. *LL* sub-band of each image *(*$LL', LL'_{noise} LL'_{crop} LL'_{rotate} LL'_{blur} LL'_{sharp}$*)* is divided into non-overlapping 4×4 coefficient blocks.
3. Apply *SVD* on each 4×4 block to get the singular values of each block $\emptyset_{ij} = (\emptyset_1, \emptyset_2, \emptyset_3, \emptyset_4)$
4. Take each samples label according to following equation
$$slabel_{ij} = \begin{cases} 1\ if\ W_{ij} = 1 \\ 2\ if\ W_{ij} = 0 \end{cases} \quad (25)$$
5. Construct the training set *TS* with feature vector $\emptyset_{ij}$ and sample label $slabel_{ij}$.
$TS_{ij}^k = (\emptyset_{ij}, slabel_{ij}), k = 1,2, \cdots, 6$ and $i \leq N, j \leq N$

Train the OSELM with the training set TS_{ij}^k

C. *Watermark Extraction*

The block diagram to extract the watermark is shown in Figure 3.

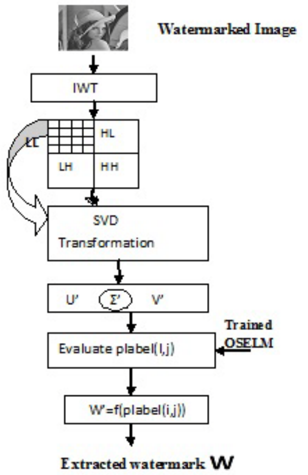


Figure 3. Proposed watermark extraction process.

The process is asfollows:

1. 1-level IWT is applied to watermarked image *C'* to get *LL'*sub-band.
2. After partitioning the *LL*sub-band into 4×4 non-overlapping coefficient blocks, apply *SVD* to each LL'_{ij} block to get the feature vector $\emptyset'_{ij} = (\emptyset'_1, \emptyset'_2, \emptyset'_3, \emptyset'_4)$
3. Get the predicted label $plabel'_{ij}$ by using the trained OSELM.
4. Watermark can be extracted by using the predicted label as:

$$W'_{ij} = \begin{cases} 1 \ if \ plabel'_{ij} = 1 \\ 0 \ if \ plabel'_{ij} = 2 \end{cases} \quad (26)$$

IV. Experimental Results for Robustness of the Proposed Watermarking Scheme

Experiments by using the proposed algorithm are performed on host images like Lena, Baboon, Pepper, Elaine and Jet of size 512×512 and a logo of size 32×32 is taken as a binary watermark. The value of embedding strength α is tested on many values and it is taken as 0.3 in our proposed algorithm.

PSNR and BER are two parameters to measure the performance of any watermarking scheme. PSNR is used to measure the imperceptibility i.e. to measure the similarity between host image and the watermarked image. The formula to calculate PSNR between two images is given by [23]

$$PSNR = 10log \log_{10} \frac{255x255}{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(C_{ij} - C'_{ij}\right)^2} /(mxn) \quad (27)$$

C and *C'* *s*are the original cover image and watermarked image. BER is used to measure the robustness of the watermarking scheme. This is used to measure the similarity between the extracted watermark from the image to the original one. The formula to calculate BER is:

$$BER = \sum_{t=1}^{pxq} (W_t \oplus W'_t)/pxq \quad (28)$$

Where W_t is original watermark and W'_t is the extracted watermark. $p \times q$ is the size of watermark and $\oplus$ is an exclusive-OR operator. Lower the value of BER implies greater similarity between the extracted watermark and the original one.

The relation between PSNR and the embedding strength α is given as:

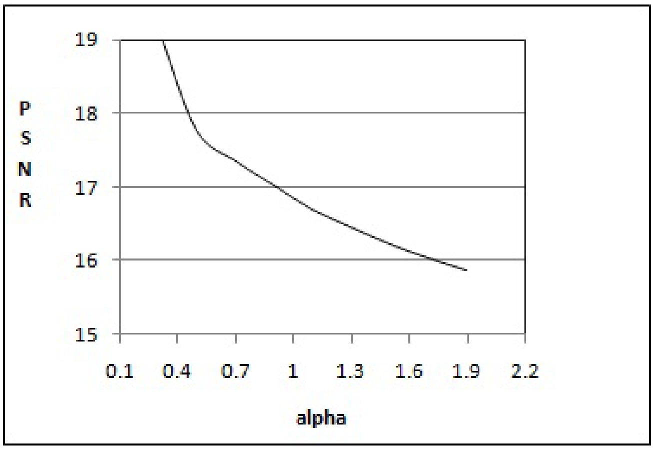


Figure 4. Relation between PSNR and α

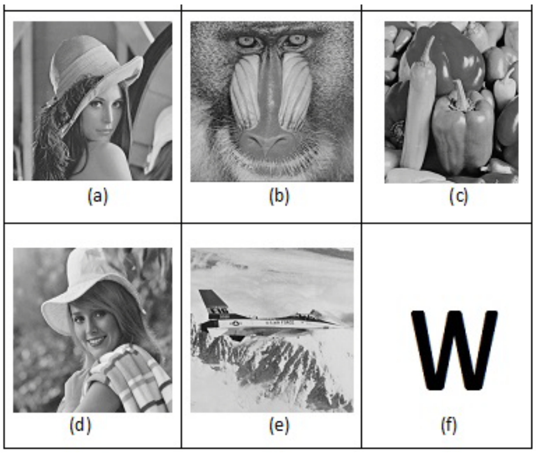


Figure 5. (a)Lena,(b)Baboon,(c)Peppers,(d)Elaine and (e)Jetplane images used for watermarking and (f) the watermark image.

Table I : The comparison results (PSNR) of *Lena* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***	***Chang et al.*[24]**	***Chen et al.* [25]**
Blurring	**23.06**	**24.14**	**29.36**	**27.75**
Cropping	**21.70**	**22.74**	**11.68**	**17.27**
Noise	**18.70**	**19.57**	**30.29**	**28.78**
Rotation	**17.49**	**17.98**	**14.95**	**15.80**
Sharpening	**18.01**	**18.01**	**27.83**	**33.12**

Table II : The comparison results (PSNR) of *Baboon* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***	***Chang et al.*[24]**	***Chen et al.* [25]**
Blurring	**19.36**	**22.98**	**21.40**	**25.50**
Cropping	**21.35**	**21.91**	**11.35**	**15.74**
Noise	**18.70**	**18.97**	**30.32**	**33.10**
Rotation	**18.29**	**16.38**	**13.83**	**14.48**
Sharpening	**14.23**	**16.53**	**17.54**	**24.60**

Table III : The comparison results (PSNR) of *Pepper* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***	***Chang et al.*[24]**	***Chen et al.* [25]**
Blurring	**27.62**	**29.16**	**28.61**	**30.70**
Cropping	**27.20**	**28.35**	**11.68**	**16.84**
Noise	**23.53**	**27.12**	**30.47**	**32.04**
Rotation	**21.59**	**22.90**	**14.82**	**17.13**
Sharpening	**23.62**	**25.77**	**26.41**	**32.52**

Table IV : The comparison results (PSNR) of *Elaine* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	**27.62**	**28.12**
Cropping	**27.20**	**27.12**
Noise	**23.53**	**24.21**
Rotation	**21.59**	**22.92**
Sharpening	**23.62**	**24.12**

Table V : The comparison results (PSNR) of *Jet* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	**27.62**	**26.12**
Cropping	**27.15**	**27.25**
Noise	**23.84**	**24.15**
Rotation	**22.12**	**22.56**
Sharpening	**23.84**	**24.94**

Table VI : The comparison results (BER) of *Lena* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***	***Shen*[26]**	***Kutter et al.* [27]**	***Yu et al.*[28]**
Blurring	0.0205	0.0119	0.0	0.0537	0.0113
Cropping	0.1074	0.1004	0.0605	0.08	0.07
Noise	0.0218	0.0218	0.0350	0.0737	0.0058

Rotation	0.1441	0.1056	0.0963	0.1225	0.1021
Sharpening	0.0469	0.0475	---	---	---

Table VII : The comparison results (BER) of *Baboon* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	0.0332	0.0123
Cropping	0.0850	0.1074
Noise	0.0928	0.10
Rotation	0.1001	0.0954
Sharpening	0.0329	0.0372

Table VIII : The comparison results (BER) of *Pepper* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	0.0405	0.0389
Cropping	0.0947	0.1009
Noise	0.0752	0.0459
Rotation	0.1121	0.0918
Sharpening	0.0596	0.0615

Table IX : The comparison results (BER) of *Elaine* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	0.0869	0.0716
Cropping	0.0215	0.0576
Noise	0.0113	0.0105
Rotation	0.1022	0.09
Sharpening	0.0669	0.0635

Table X : The comparison results (BER) of *Jet* image under various attacks with *ELM* and *OSELM*

Attacks	***ELM***	***OSELM***
Blurring	0.0415	0.0383
Cropping	0.1055	0.1279
Noise	0.0635	0.0756
Rotation	0.1078	0.0918
Sharpening	0.0613	0.0591

Table XI : Attacked Lena images,corresponding PSNR, BER and ownership (extracted watermark) for copyright protection with OSELM.

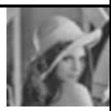
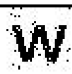
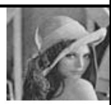
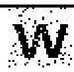
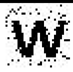
Attacks	***Image***	***BER***	***PSNR***	***Ownership***
Blurring		**0.0119**	**24.14**	
Cropping		**0.1004**	**22.74**	
Noise		**0.0218**	**19.57**	
Rotation		**0.1056**	**17.98**	
Sharpening		**0.0475**	**18.01**	

Table XII : Attacked Baboon images,corresponding PSNR, BER and ownership (extracted watermark) for copyright protection with OSELM.

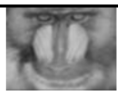
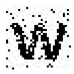
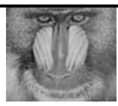
Attacks	***Image***	***BER***	***PSNR***	***Ownership***
Blurring		**0.0123**	**22.98**	
Cropping		**0.1074**	**21.91**	
Noise		**0.10**	**18.97**	
Rotation		**0.0954**	**16.38**	
Sharpening		**0.0372**	**16.53**	

Table XIII : Attacked Pepper images,corresponding PSNR, BER and ownership (extracted watermark) for copyright protection with OSELM.

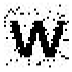
Attacks	*Image*	*BER*	*PSNR*	*Ownership*
Blurring		**0.0389**	**29.16**	
Cropping		**0.1009**	**28.35**	
Noise		**0.0459**	**27.12**	
Rotation		**0.0918**	**22.90**	
Sharpening		**0.0615**	**25.77**	

Table XIV : Attacked Elaine images,corresponding PSNR, BER and ownership (extracted watermark) for copyright protection with OSELM.

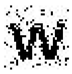
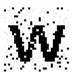
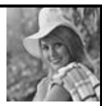
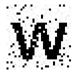
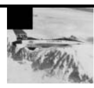
Attacks	*Image*	*BER*	*PSNR*	*Ownership*
Blurring		**0.0716**	**28.12**	
Cropping		**0.0576**	**27.12**	
Noise		**0.0105**	**24.21**	
Rotation		**0.09**	**22.92**	
Sharpening		**0.0635**	**24.12**	

Table XV : Attacked Jet images,corresponding PSNR, BER and ownership (extracted watermark) for copyright protection with OSELM.

Attacks	*Image*	*BER*	*PSNR*	*Ownership*
Blurring		**0.0383**	**26.12**	
Cropping		**0.1279**	**27.25**	
Noise		**0.0756**	**24.15**	
Rotation		**0.0918**	**22.56**	
Sharpening		**0.0591**	**24.94**	

V. CONCLUSION

In this paper, we proposed a novel combination of SVD and OSELM in IWT domain for copyright protection. In this watermarking scheme, a cover image is transformed using IWT domain to get the *LL* sub-band. This sub-band is divided into 4×4 non-overlapping coefficients and then SVD is applied to each block to get the singular values. Each bit of binary watermark logo is embedded to one block by modifying their singular values using the given numerical operations. Next phase of this scheme is to train an OSELM machine to learn the relationship between the original coefficient and their watermarked version. During the extraction phase this trained OSELM machine is used to extract the watermark from the watermarked image. Experiments have been performed on various attacked images and it is shown from the results that even after attacks, the value of BER is very less, so the extracted watermark has very much similarity with the original one, so helpful to prove the ownership of the data.

VI. REFERENCES

[1] S. Mohanty, "Digital Watermarking: A tutorial review," 1999.

[2] R. Liu and T. Tan, "A SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimed.,* vol. 4(1), p. 121–128, 2002.

[3] R. A. Ghazy, El-Fishawy N A and Hadhoud M M, "An eficient block-by-block SVD-based image watermarking scheme," in *Radio Science Conference, NRSC,* 2007.

[4] Quan Liu and Qingsong Ai, "Combination of DCT-based and SVD-based watermarking scheme," in *Signal Processing Proceedings, ICSP, 7th International Conference*, 2004.

[5] Ganic Emir and Ahmet Eskicioglu M, "Robust DWT-SVD domain image watermarking: Embedding data in all frequencies," in *Proceedings of the workshop on Multimedia and Security*, 2004.

[6] G.B.Huang, Q.Y.Zhu and C.K.Siew, "Extreme learning machine: theory and applications," *Neurocomputing,* vol. 70, pp. 489-501, 2006.

[7] A.R.Calderbank et al, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal,* vol. 5 (3), pp. 332-369, 1998.

[8] M. Ramani, Dr. E. V. Prasad and Dr. S. Varadarajan, "Steganography Using BPCS the Integer Wavelet Transformed image," *UCSNS International Journal of Computer Science and Network Security,* vol. 7, 2007.

[9] S. Lee, C.D. Yoo and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Transactions on Information Forensics and Security,* vol. 2, pp. 321-330, 2007.

[10] S.Jayasudha, "Integer Wavelet Transform Based Steganographic Method Using Opa Algorithm," *Research Inventy International Journal Of Engineering And Science,* vol. 2, pp. 31-35, Feb 2013.

[11] T, Ogden CJ and Huff, "The singular value decomposition and its application in image processing," *Lin. Algebra-Maths- 45, College of Redwoods,* 1997.

[12] McIndoo, Jody S. Hourigan and Lynn V, "Singular value decomposition," *Lin. Algebra-Maths-45, College of Redwoods,* 1998.

[13] C.L. Patterson and H. C. Andrews, "Singular value decompositions and digital image processing," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vols. ASSP-24, no. 26-53, 1976.

[14] Dobrovolny M. Silar Z. and Cerny M., "Asymmetric Image Compression for Embedded Devices based on Singular Value Decomposition," *IEEE Applied Electronics Pilsen,* 2011.

[15] S.K. Singh, S. Kumar and Athens, "A Framework to Design Novel SVD Based Color Image Compression. s.n., 2010," *Computer Modeling and Simulation, 2009. EMS '09.*

[16] A. Shnayderman, A. Gusev and A. M. Eskicioglu., "A Multidimensional Image Quality Measure Using Singular Value Decomposition," *IS&T SPIE Symposium on Electronic Imaging 2004, Image Quality and System Performance.,* pp. 18-22, 2004.

[17] Zhou B and Chen J, "A geometric distortion resilient image watermarking algorithm based on SVD.," *Chin. J. Image Graphics.,* vol. 9(1), p. 506–512, 2004.

[18] C.R.Rao and S.K.Mitra, "Generalized Inverse of Matrices and its Applications," *Wiley,* 1971.

[19] G-B. Huang, D-H. Wang and Y. Lan, "Extreme Learning Machine : A Survey," *International Journal of Machine Learning and Cybernitics,* vol. 2, pp. 107-122, 2011.

[20] G-B Huang, L. Chen L and C-K.Siew, "Extreme learning machine: a new learning scheme of Feedforward networks," vol. 2, pp. 985-990, 2004.

[21] P.L. Bartlett, "The sample complexity of patten classification with neural networks:The size of weights is more important than the size of the network," *IEEE Trans. of Information Theory,* vol. 44, pp. 525-536, 2006.

[22] N.Y. Liang, G.B. Huang, P. Saratchandran and N. Sundarajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. Neural Netw,* pp. 1411-1423, 2006.

[23] Ben Wang, Jinkou Ding, Qiaoyan Wen, Xin Liao and Cuixiang Liu, "An Image Watermarking Algorithm Based On DWT DCT and SVD," in *Proceedings of IC-NIDC,* 2009.

[24] C.Y. Chang, H.J. Wang and S.W. Pan, "A robust DWT-based copyright verification scheme with Fuzzy ART," *J. Syst. Softw.,* vol. 82, pp. 1906-1915, 2009.

[25] T.H.Chen, G.Horng and W.B.Lee, "A publicly verifiable copyright-proving scheme resistant to mallicious attacks," *IEEE Trans. Ind. Electron,* vol. 51(1), pp. 327-334, 2005.

[26] R. Shen, Y.G. Fu and H.T. Lu, "A novel image watermarking scheme based on support vector regression," *J. Syst. Softw.,* vol. 78, pp. 1-8, 2005.

[27] M. Kutter, F. Jordan and F. Bossen, "Digital signature of color images using amplitude modulation," *J. Electron. Imag.,* vol. 7 (2), pp. 326-332, 1998.

[28] P.T. Yu, H.H. Tsai and D.W. Sun, "Digital watermarking based on neural networks for color images," *Signal Process.,* vol. 81 (3), p. 663–671, 2001.

