A Novel Blind Digital Watermarking Based on SVD and Extreme Learning Machine

NEELAM DABAS¹, RAMPAL SINGH² and VIKASH CHAUDHARY³

¹Computer Science Department, University of Delhi, Delhi, India.
²Computer Science Department, DDUC University of Delhi, Delhi, India.
³Computer Science Department, BNC University of Delhi, Delhi, India.
*Corresponding author E-mail: ansingh.du@gmail.com

http://dx.doi.org/10.13005/ojcst/10.01.21

(Received: March 05, 2017; Accepted: March 16, 2017)

ABSTRACT

Modification of media and illegal production is a big problem now a days because of free availability of digital media. Protection and securing the digital data is a challenge. An Integer Wavelet Transformation (Wt) domain based robust watermarking scheme with Singular Value Decomposition (SVD) and Extreme Learning Machine (ELM) have been proposed and tested on different images. In this proposed scheme, a watermark or logo is embedded in the Wt domain as ownership information with SVD and ELM is trained to learn the relationship between the original coefficient and the watermarked one. This trained ELM is used in the extraction process to extract the embedded logo from the image. Experimental results show that the proposed watermarking scheme is robust against various image attacks like Blurring, Noise, Cropping, Rotation, Sharpening etc. Performance analysis of proposed watermarking scheme is measured with Peak Signal to Noise Ratio (PSNR) and Bit Error Rate (BER).

Keywords: Wt, SVD, ELM, PSNR, BER.

INTRODUCTION

With the invention and expansion of internet, data in digital form is distributed and copied easily worldwide. But with the distribution, protection or security of data is equally important. Watermarking is an emerging technique for the security of data. Watermarking is the process that embeds data called a watermark, tag or label into a multimedia object such that watermark can be detected or extracted later to prove the ownership of ownership etc¹. Over the past years, many singular value decomposition (SVD) based watermarking schemes are proposed²,³,⁴, in which three matrices are modified slightly to embed the watermark. Later, all these SVD based watermarking algorithms are extended to embed the watermark in wavelet domains to provide better robustness⁵. We are proposing a method with the combination of SVD and Extreme Learning Machine (ELM) in Integer Wavelet Domain (Wt). ELM is an algorithm for single layer feed forward neural network, where parameters of neural network like weights and bias are randomly selected. Training time of ELM is very
fast since weights and bias are not adjusted by using gradient descent method\(^6\). Gradient descent method have the problem of slow learning rate, local minima etc. IWT domain reduces the signal loss during the inverse process.

Rest of the paper is organized as follows. We give some background theories about IWT, SVD and ELM in Section 2. Proposed water marking schemes watermark embedding, ELM training and watermark extraction are described in Section 3. Experimental results are discussed in Section 4 followed by conclusion in Section 5.

**Literature Survey Of IWT, SVD And ELM**

**Integer Wavelet Transform (IWT)**

To increase the robustness, watermarks are to be embedded in wavelet domain instead of spatial domain. The image is divided into low and high resolution bands \((LL, HL, LH, HH)\). In discrete wavelet transform, we hide data into floating point coefficients, so during the inverse transformation, any truncation in floating point value leads to the loss of information. IWT transforms a data set into another integer data set\(^7\). So during forward and inverse transformation, no loss of information is there which leads to have a very close copy of original image\(^8\). Lifting schemes are used to perform IWT. IWT process is divided into three steps\(^9\).

1. **Split**: Partition the data set \(\beta\) into low and high frequency samples.
   
   \[ \text{Split } (\beta, \beta) = \{ \text{odd } (\beta, \beta) \} = \{ \gamma, \gamma \} \]

2. **Predict**: Predict the odd elements \(\gamma\) from the even elements \(\lambda\).

3. **Update**: Update the data in the set with the data in the set.

**Singular Value Decomposition (SVD)**

In linear algebra, SVD is a technique to factorize a rectangular matrix into three decomposition matrices

\[
\begin{align*}
\mathbf{I} & = \mathbf{U} \mathbf{D} \mathbf{V}^T \\
\mathbf{U} & = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \\
\mathbf{D} & = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_n \end{pmatrix} \\
\mathbf{V} & = [v_1, v_2, \ldots, v_n] \\
\end{align*}
\]

By multiply \(\mathbf{U}, \mathbf{D}, \mathbf{V}^T\), we can get the matrix \(\mathbf{I}\).

\[
\mathbf{I} = \mathbf{U} \mathbf{D} \mathbf{V}^T 
\]

**Extreme Learning Machine (ELM)**

ELM algorithm is proposed by Huang \(\text{et al.}^{12}\) which is based on SLFNs. ELM overcomes the pitfalls of Artificial Neural Network like local minima and slow learning rate. Consider a SLFN with \(N\) input layer, \(M\) output layer with \(L\) hidden neuron. Take \(N\) samples where \(\mathbf{x}\) is vector and \(\mathbf{t}\) is \(\mathbf{M}\) output vector. Randomly select two parameters, bias and weights where \(\mathbf{w}\) is the weight vector of the connection between \(i\)th input layer and \(k\)th hidden neuron of input layer. The output function \(T\) with activation function is given by

\[
T(x) = \sum_{i=1}^{L} \mathbf{\beta}_i \mathbf{g}_i(x) = \sum_{i=1}^{L} \mathbf{\beta}_i G(w_i, b_i, x) = \mathbf{a}_i 
\]

Where \(\mathbf{w}\) is the weight vector connecting the \(i\)th hidden node and the output nodes, \(b_i\) is the threshold value of \(i\)th hidden nodes. For additive hidden nodes the activation function is defined as

\[
G(w_i, b_i, x) = g(w_i \cdot x + b_i) 
\]

Where denotes the inner product of vector and \(x\). For an RBF hidden node the activation function \(g(x)\) is given by

\[
G(w_i, b_i, x) = g(|| x - w_i ||) 
\]

Where \(w_i\) and \(b_i\) are the center and the impact factor of the \(i\)th RBF node. Equaion (4) can be re-written as

\[
\mathbf{H} \mathbf{\beta} = \mathbf{T} 
\]
Where \( H \) is the hidden layer output matrix and \( T \) is the target vector. are estimated as
\[
\beta = H^\dagger T \tag{9}
\]
where \( H^\dagger \) is the Moore-Penrose generalized pseudo inverse\(^{13}\).

**Essence of ELM**

The basic essence of ELM is that:
1) No iterative tuning is required in SLFN,, parameters of hidden layer are randomly chosen\(^{12,14}\).
2) \( ||H\beta - T|| \), the training error and \( \| \| \) norm of output weight need to be minimized\(^{12,14,15}\).
3) Least square method is used to calculate between the hidden layer and the output layer\(^{16,6}\).

**Algorithm**

ELM algorithm is as follows: For \( N \) training samples \((x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m, L \) number of hidden neurons and \( g(x) \) as an activation function
1) Input weight \( W_i \) and bias \( b_i \) are randomly generated, where \( i=1,...,N \)
2) Calculate \( H \), hidden neuron output matrix.
3) Calculate \( \beta \), output weight using equation
\[
\beta = H^\dagger T
\]

**Proposed Watermarking Scheme**

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

**Watermark Embedding**

The block diagram of watermark embedding is shown in Fig(1).

Let us assume there is a host image \( H \), of size and a binary watermark image \( w \) with size. As it is a binary image so . In our case, size of image \( H \)s and watermark \( w \)s. The embedding algorithm is as follows:

1. Host image \( H \) is transferred through 1-level IWT to decomposed it into \((LL, LH, HL, HH)\) sub-bands.
2. Out of these four sub-bands \((LL, LH, HL, HH)\), LL (Lowest level) has been selected for watermark embedding as it contains maximum energy. Now \( LL \) sub-band is partitioned into \( 4 \times 4 \) non-overlapping coefficient blocks.
3. Apply SVD on each blocks to get three components, \( U, D \) and \( V \)
\[
LL = U D V^T \tag{9}
\]
4. Take out the four singular values of each \( LL \)
\[
D = (d_1, d_2, d_3, d_4)
\]
5. Modify the values of $d_2$ based on the following mathematical operations where $\alpha$ is embedding strength of watermarking scheme

$$
\begin{align*}
    d_2 &= \begin{cases} 
    \mu + \alpha \times \delta, & \text{if } w_{ij} = 1 \text{ and } d_2 < \mu \\
    d_2, & \text{if } w_{ij} = 1 \text{ and } d_2 > \mu \\
    \mu - \alpha \times \delta, & \text{if } w_{ij} = 0 \text{ and } d_2 > \mu \\
    d_2, & \text{if } w_{ij} = 0 \text{ and } d_2 < \mu 
    \end{cases} \\
    \mu &= \frac{1}{4} \times \sum_{i=1}^{4} d_i \\
    \delta &= \frac{1}{4} \times \sum_{i=1}^{4} (d_i - \mu)^2 
\end{align*}
$$

(10)

6. Perform inverse SVD to get the modified coefficient block.

$$
U_i D_i' V_i = LL_i' 
$$

(11)

7. Perform inverse IWT with $LL_i' HL, LH, HH$ to get the final watermarked image $H'$.

ELM Training

The block diagram to train an ELM is shown in fig (2).

Training set is derived from the original watermarked image $H'$ and the corrupted watermarked images by Gaussian noise $H'_\text{noise}$, cropping $H'_\text{crop}$, rotation $H'_\text{rotate}$, blurring $H'_\text{blur}$, sharpening $H'_\text{sharp}$ respectively. Training of ELM is as follows:

1. Apply IWT on $H'$, $H'_\text{noise}$, $H'_\text{crop}$, $H'_\text{rotate}$, $H'_\text{blur}$, $H'_\text{sharp}$ and get their respective $LL$ sub-bands denoted as $LL', LL'_\text{noise}, LL'_\text{crop}, LL'_\text{rotate}, LL'_\text{blur}, LL'_\text{sharp}$.
2. Each $LL$ sub-band ($LL', LL'_\text{noise}, LL'_\text{crop}, LL'_\text{rotate}, LL'_\text{blur}, LL'_\text{sharp}$) is divided into non-overlapping coefficient blocks.
3. Apply SVD on each block to get the singular values of each image $D_i = (d_1^i, d_2^i, d_3^i, d_4^i)$.
4. Take each sample label according to the following equation

$$
\text{label}_{ij} = \begin{cases} 
    1 \text{ if } W_{ij} = 1 \\
    2 \text{ if } W_{ij} = 0 
\end{cases} 
$$

(12)

5. Construct the training set with feature vector $D_i$ and label $\text{label}_i$

$$
T_i = (D_i, \text{label}_i), i=1,..,6 \text{ and } i\leq N \text{ and } j\leq N 
$$

6. Train the ELM with the training set $T_i$.

Watermark Extraction

The block diagram to extract the watermark is shown in fig (3).

The process is as follows:

1. Apply IWT on watermarked image $H'$ to get $LL'$ sub-band.
2. Partition $LL'$ sub-band into $4 \times 4$ non-overlapping coefficient blocks.
3. Apply SVD on $LL'_i$ to get the feature vector $D_i' = (d_1^i, d_2^i, d_3^i, d_4^i)$.
4. By using well trained ELM, get the predicted label $\text{label}'_{ij}$ corresponding to each $D_i'$.
5. Watermark can be extracted by using the predicted label as

$$
W_i' = \begin{cases} 
    1 \text{ if } \text{label}'_{ij} = 1 \\
    0 \text{ if } \text{label}'_{ij} = 2 
\end{cases} 
$$

(13)

Experimental Results For Robustness Of The Proposed Watermarking Scheme

In this paper, an experiment is performed on host images like Lena, Baboon, Pepper, Elaine and Jet of size 512×512 and a watermark logo of size 32×32 is used. The value of $\alpha$ is taken as 0.3. Performance of watermarking algorithm is done on the basis of two parameters imperceptibility and robustness. PSNR is used to measure the quality of watermarked image with the original host image. Higher the value of PSNR, better is the quality of watermarked images. PSNR between the original image and watermarked image $H$ and $H'$ is $17$. 

![Watermark Extraction Diagram](image-url)
and BER is evaluated as

\[
BER = \frac{\sum_{i=1}^{pxq} (w_i \oplus w'_i)}{pxq}
\]

where \( w_i \) is original watermark and \( w'_i \) is the extracted watermark. \( pxq \) 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 value of \( \alpha \) is tested in the interval \( \alpha \in [0,0.3] \). It is found that the value of PSNR is decreasing with the value of \( \alpha \) as shown in Fig(4).

### Table 1: Attacked Lena images, corresponding value of BER, PSNR and extracted binary watermark (ownership) from attacked watermarked Lena image by using the proposed watermarking scheme

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Image</th>
<th>BER</th>
<th>PSNR</th>
<th>Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring</td>
<td><img src="image.png" alt="Image" /></td>
<td>0.0205</td>
<td>23.06</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>Cropping</td>
<td><img src="image.png" alt="Image" /></td>
<td>0.1074</td>
<td>21.70</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>Noise</td>
<td><img src="image.png" alt="Image" /></td>
<td>0.0218</td>
<td>18.7039</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>Rotation</td>
<td><img src="image.png" alt="Image" /></td>
<td>0.1441</td>
<td>17.4914</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
<tr>
<td>Sharpening</td>
<td><img src="image.png" alt="Image" /></td>
<td>0.0469</td>
<td>18.0145</td>
<td><img src="image.png" alt="Image" /></td>
</tr>
</tbody>
</table>
Table 2: Attacked Baboon images, corresponding value of BER, PSNR and extracted binary watermark (ownership) from attacked watermarked Baboon image by using the proposed watermarking scheme

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Image</th>
<th>BER</th>
<th>PSNR</th>
<th>Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring</td>
<td>![Blurring Image]</td>
<td>0.0332</td>
<td>19.3670</td>
<td>![W]</td>
</tr>
<tr>
<td>Cropping</td>
<td>![Cropping Image]</td>
<td>0.0850</td>
<td>21.3491</td>
<td>![W W W]</td>
</tr>
<tr>
<td>Noise</td>
<td>![Noise Image]</td>
<td>0.0928</td>
<td>18.2904</td>
<td>![W W W W]</td>
</tr>
<tr>
<td>Rotation</td>
<td>![Rotation Image]</td>
<td>0.1001</td>
<td>15.3847</td>
<td>![W W W]</td>
</tr>
<tr>
<td>Sharpening</td>
<td>![Sharpening Image]</td>
<td>0.0329</td>
<td>14.2370</td>
<td>![W W]</td>
</tr>
</tbody>
</table>

Table 3: Attacked Peppers images, corresponding value of BER, PSNR and extracted binary watermark (ownership) from attacked watermarked Peppers image by using the proposed watermarking scheme

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Image</th>
<th>BER</th>
<th>PSNR</th>
<th>Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring</td>
<td>![Blurring Image]</td>
<td>0.0405</td>
<td>27.6210</td>
<td>![W]</td>
</tr>
<tr>
<td>Cropping</td>
<td>![Cropping Image]</td>
<td>0.0947</td>
<td>27.2009</td>
<td>![W W]</td>
</tr>
<tr>
<td>Noise</td>
<td>![Noise Image]</td>
<td>0.0752</td>
<td>23.5349</td>
<td>![W W W]</td>
</tr>
<tr>
<td>Rotation</td>
<td>![Rotation Image]</td>
<td>0.1121</td>
<td>21.5885</td>
<td>![W W]</td>
</tr>
<tr>
<td>Sharpening</td>
<td>![Sharpening Image]</td>
<td>0.0596</td>
<td>23.6203</td>
<td>![W W]</td>
</tr>
</tbody>
</table>
Table 4: Attacked Elaine images, corresponding value of BER, PSNR and extracted binary watermark (ownership) from attacked watermarked Elaine image by using the proposed watermarking scheme

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Image</th>
<th>BER</th>
<th>PSNR</th>
<th>Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring</td>
<td>![Image]</td>
<td>0.0869</td>
<td>23.8083</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Cropping</td>
<td>![Image]</td>
<td>0.0215</td>
<td>17.9941</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Noise</td>
<td>![Image]</td>
<td>0.0113</td>
<td>16.9005</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Rotation</td>
<td>![Image]</td>
<td>0.1022</td>
<td>16.1664</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Sharpening</td>
<td>![Image]</td>
<td>0.0669</td>
<td>15.1334</td>
<td>![Watermark]</td>
</tr>
</tbody>
</table>

Table 5: Attacked Jetplane images, corresponding value of BER, PSNR and extracted binary watermark (ownership) from attacked watermarked Jetplane image by using the proposed watermarking scheme

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Image</th>
<th>BER</th>
<th>PSNR</th>
<th>Ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blurring</td>
<td>![Image]</td>
<td>0.0415</td>
<td>25.9971</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Cropping</td>
<td>![Image]</td>
<td>0.1055</td>
<td>25.6287</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Noise</td>
<td>![Image]</td>
<td>0.0635</td>
<td>23.0474</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Rotation</td>
<td>![Image]</td>
<td>0.1078</td>
<td>20.1516</td>
<td>![Watermark]</td>
</tr>
<tr>
<td>Sharpening</td>
<td>![Image]</td>
<td>0.0613</td>
<td>23.2219</td>
<td>![Watermark]</td>
</tr>
</tbody>
</table>
CONCLUSIONS

In this paper, we proposed a novel combination of IWT, SVD and ELM for authentication of ownership. In the proposed scheme, host image is transformed in IWT domain and then LL sub-band is used to take the singular values, where required numerical operations are done to embed the watermark. Watermark extraction is a two-step process, firstly training of ELM and secondly the actual watermark extraction for proof of ownership. As shown in experimental results, our proposed method is robust against various attacks and the extracted watermark, to prove the ownership is very much similar to the original watermark, i.e. less BER value.

REFERENCES