Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science 281 (2011) 35–50

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Neural Watermark Approach

Jose Aguilar[1,2]

Departamento de Computacin
Universidad de Los Andes
Mrida, Venezuela

Juan Anderson

Instituto IESA Caracas, Venezuela

Abstract

In this paper we propose the coupling of a watermarking technique for images, called least significant bit, in the multiple classes random neural network. For that, we design a training process of the watermark pattern, an embedding process of the learned pattern in the original image, and a detecting process of this pattern in the carrier image. The removal of the watermark is not considered in this work, since the aim is to study the capability of detection of our neural approach of any manipulation over the carrier image. We define several attacks to compare the robustness of our approach with previous works (rotation, JPEG compression, scaling, noise, cropping, Horizontal Flip, Brightness and contrast correction), obtaining very good results with our approach. Additionally, we obtain very good performances in terms of the Peak Signal to Noise Ratio and Noise Generated criteria.

Keywords: Digital watermark, Random Neural Network, LSB Technique, Image Violation Detection Process.

1 Introduction

Clearly, the boom of Internet and Distributed Systems concerning the exchange of information, have allowed that the information to be affordable for everyone. Due to that, digital security has become today in an area of great importance. The problem arises when the content of information must be protected in the processes of communication, finding in an ocean of possible attacks on this information. In these cases we require digital security techniques to protect the contents of the information. The digital watermarking is one of the techniques used to protect the information using a mark that is embedded in a digital object (image, sound, video

[1] Thanks to the CDCHT Project I-1237-10-02-AA of the Universidad de los Andes for the financial support

[2] Email: aguilar@ula.ve

doi:10.1016/j.entcs.2011.11.024

or text), where its main function is to reveal whether the digital content has been violated or not.

The digital watermarks have been a subject of study for several years, and there are several techniques and methods. Particularly, we study in this work the image digital watermarking. Some of the recent works on watermarking are: in [4] is presented an overview of watermarking techniques appeared in the literature during the last five years. Additionally, they give a general classification of watermarking algorithms on the basis of their embedding and decoding characteristics. In their approach multiplies the watermark with the transformed source, instead of adding the watermark to the source image, as is normally done. In [11] is introduced an algorithm based on Discrete Cosine Transform (DCT) and Discrete Wavelet Transforms (DWT). In this algorithm, the information of digital watermarking which has been discrete Cosine transformed, is put into the high frequency band of the image which has been wavelet transformed. Then, they distill the digital watermarking with the help of the original image and the watermarking image. [5] presents an algorithm for digital image watermarking based on Singular Value Decomposition (SVD), Multiple Descriptions (MD) and Quantization Index Modulation (QIM) of the host image. Watermark embedding is done at two stages. In the first stage, DCT of odd description of the host image is computed. In the second stage, a copy of the watermark image is embedded in the watermarked image generated at the first stage. In [7] suggests a watermarking technique that uses artificial immune recognition system to protect color image's intellectual property rights. The watermark is embedded in the blue channel of a color image. m-bit binary sequence embedded into the color image is used to train artificial immune recognition system. In [14] they introduce the notion of Video Watermarking and features required to design a robust watermarked video. [2] proposes an approach which uses the Lifting wavelet transform for decomposition of the original image. Some recent works on watermarking approaches based on artificial neural networks are: [12] develops a SVD based image watermarking, an error control coding (ECC) and an artificial neural networks. Specifically, the artificial neural networks are used for the authentication process to increase the robustness of the method against malicious attacks. The purpose of [13] is to study the performance of an approach for digital watermarking that uses neural networks in terms of its capacity, transparency, and robustness.

This paper propose the use of the random neural networks model to generate and detect digital watermark in images, in order to exploit the highly efficient learning and pattern recognition mechanisms of this model. Particularly, we propose the mixing of a watermarking technique in images called least significant bit, or LSB with the multiple classes random neural network model (MCRNN). The LSB technique exploits the weakness that the human visual system HVS has to detect brightness variations in the colors of the visual spectrum [6]. The technique is based on replacing the least significant bit of the carrier signal with the bit pattern of the digital watermark. The detection process can recover the watermark if it has known the values used to embed the digital watermark. The watermark embedding process is performed replacing the last bit of each pixel of the original image, according to a pattern of a pre-selected secret key [3]. This technique can detect any changes in the image, but is little robust because if the digital watermark is detected the original image can be violated.

We attack this problem encrypting the watermark in the synapses weights of the MCNN (it learns the watermark) and proposing a neural-LSB embedded and detection approach. The watermark is embedded and detected through a MCRNN, which is learned by the MCRNN in the training process. In our approach based on the MCRNN and on the LSB technique, the MCRNN learns a version of the watermark built using the LSB principle (the synapses weights will contain the encrypted watermark), and uses the synapses weights of the MCRNN in the neural-LSB embedded and detection phases. Particularly, the detection of a violation in the original image is like a classical recognition problem on the domain of Artificial Neural Network (in our case, of the encrypted watermark embedded in the original image).

Our proposed method is highly robust; the watermarked image can survive to different image attacks like rotation, JPEG compression, scaling, noise, cropping, Horizontal Flip, Brightness and contrast correction; contrary to the classical LSB technique, due to that we encrypt the watermark (in the synapses weights of the MCNN) and we use it in the watermarking process (embed and detection process). The watermark is difficult to violate and the watermarking process is very sensitive to a modification of the original image (it does not recognize the watermark if this one is modified). Additionally, our approach is very good in terms of Peak Signal to Noise Ratio (PSNR) and Noise Generated (NG) criteria, which makes invisible our approach (it is not perceived in the image).

The rest of the paper is organized as follows. Section 2 introduces the MCNN. Section 3 contains the watermarking method proposed in this work. Section 4 contains the experiments and results analysis. Section 5 contains the conclusions

2 The multiple classes random neural network

The MRNN is based on the random neural network model. The RNN was introduced by Erol Gelenbe [9. 10], it consists of a network of n neurons in which positive and negative signals circulate. Each neuron accumulates signals as soon as they arrive, and can fire if the count of signals in a given moment is positive. The firing occurs randomly according to an exponential distribution of constant rate, and the signals are sent to other neurons or from outside the network. Each neuron i of the network is represented at any time t by the potential of its input signal $k_i(t)$. The positive and negative signals have different roles in the network. A negative signal reduces by 1 the potential of the neuron to which it arrives (inhibition) or no effect on the potential if it is zero, while a positive signal increases by 1 the potential of the neuron. The signals can arrive at a neuron from outside the network or from other neurons. Each time a neuron fires is deleted the potential contained in it. A signal leaves the neuron i due to the neuron j with probability $p^+(i, j)$ as a positive signal (excitation) or as negative signal with probability $p^-(i,j)$ (inhibition), or leaves the network with probability d(i). Positive signals arrive outside the ith-neuron according to a Poisson process of rate $\Lambda(i)$ (external excitation signals), and external negative signals arrive to the ith-neuron according to a Poisson process of rate $\lambda(i)$ (external inhibition signals). The rate at which neuron i fires is r(s). The main property of this model is the probability of excitation of a neuron i, q(i) [9, 10].

The MCNN is an extension of the RNN model [8], where the positive signals may belong to many classes and the potential in a neuron is represented by a vector $K_i = (K_{i1}, ..., ..., K_{iC})$, where K_{ic} is the value of "class c potential" of neuron i, or its "excitation level in terms of class c signals", and the negative signals only belong to one class. The total potential of neuron i is [8]:

$$K_i = \sum_{c=1}^{C} K_{ic} \tag{1}$$

When a positive signal of class c arrives at a neuron simply K_{ic} increases by 1, and when a negative signal reaches it, if $K_i > 0$, the potential is reduced by 1, and the kind of potential to be reduced is chosen randomly with probability K_{ic}/K_i for any c = 1, ..., C. A negative signal reaches a neuron that has the potential zero has no effect. Exogenous positive signals of class c arrive at a neuron i with a Poisson stream of rate $\Lambda(i,c)$, while exogenous negative signals arrive with a Poisson stream of rate $\lambda(i)$. A neuron is excited if its potential is positive. Then, the neuron fires at intervals exponentially distributed sending excitation signals of different classes, or inhibitory signals to other neurons or outside the network. Thus, the neuron i can fire when its potential is positive ($K_i > 0$). The neuron i sends excitation signals of class c at rate $r(i,c) > 0$, with probability K_{ic}/K_i. A signal left class c of neuron i to the class ϕ of neuron j with probability $p^+(i,c;j,\phi)$ as a positive signal (excitation) or as negative signal with probability $p^-(i,c;j)$ (inhibition). As is defined in [8], we have:

$$\sum_{j,\phi} p^+(i,c;j,\phi) + \sum_{j} p^-(i,c;j) + d(i,c) = 1 \quad \forall i = 1,n \quad and \quad c = 1,C \tag{2}$$

Let K(t) the vector representing the state of the neural network in a time t and let $K = (K_1, ..., K_n)$ a particular value of the vector. The main property of this model is the probability of excitation of a "class ϕ" of the neuron j, q(j, ϕ), with $0 < q(j,\phi) < 1$, which satisfies the nonlinear equation [8]:

$$q(j,\phi) = \frac{\lambda^+(j,\phi)}{r(j,\phi) + \lambda^-(j)} \tag{3}$$

where,

$$\lambda^+(j,\phi) = \sum_{i,c} q(i,c)r(i,c)p^+(i,c;j,\phi) + \Lambda(j,\phi) \tag{4}$$

$$\lambda^-(j) = \sum_{i,c} q(i,c)r(i,c)p^-(i,c;j) + \lambda(j) \tag{5}$$

The synaptic weights for positive ($w^+(i,c;j,\phi)$) and negative ($w^-(i,c;j)$) signals are defined as:

$$w^+(i,c;j,\phi) = r(i,c)p^+(i,c;j,\phi) \tag{6}$$

$$w^-(i,c;j) = r(i,c)p^-(i,c;j) \tag{7}$$

and if d(i, c) = 0, the firing rate r(i, c) is:

$$r(i,c) = \sum_{j,\phi} w^+(i,c;j,\phi) + \sum_{j} w^-(i,c;j) \tag{8}$$

3 Our Hybrid LSB-MCRNN Watermarking Technique

Our approach mixes the LSB technique in the MCRNN model. To explain our approach, we are going to use an example, suppose 2 images of 24-bit depth, so the pixels in each image will be configured for 3 channels of 8 bits each, representing the RGB (Red, Green and Blue) format, a channel for red, other channel for green and the last channel for blue. These three channels cover a large range of colors, ranging from black (00000000) to white (11111111). The first image is the original image, and the second image is the pattern to embed. The combination of the LSB technique and the MCRNN consist on taking the three least significant bits of each pixel of the watermark (one for each channel) as the 3 classes of each neuron in the network (three classes per neuron, and for the value of each class we use the LSB approach in each pixel). This MCRNN will be used in the embedding and detecting digital watermark processes.

The MCRNN is configured like a single layer of neurons fully interconnected such that each neuron N(i) is associated with each pixel of the image pattern of the watermark, denoted by a row and column (f, c) according to their position in the binary matrix. Each neuron N(i) contain 3 classes directly related to the least significant bit of each channel of each pixel (for example, a neuron with the following values N(1) = (1,0,1) means that the pixel number 1 has the values 1, 0 and 1 as the values of the least significant bits of the three RGB channels). Now, the MCRNN learns this pattern of the watermark, so that later it can be used in the embedding and detection process of our watermarking method.

In general, our approach is composed by three processes: a training process of the pattern of the watermark, an embedding process of the pattern in the original image, and a detecting process of this pattern in the carrier image.

3.1 Training process

Our training stage can be regarded as a process to encrypt the watermark for the embedding and detection processes. In this section we explain the main aspect of this phase (meaning of the weights, initialization of certain parameters of the MCRNN, learning procedure, etc.).

3.1.1 Synaptic weights, inputs and outputs

As is known, in the synaptic weights is stored the knowledge learned by the network. We need a given set of input and output pairs (X, Y) to train the positive

$W^+(i,c;j,z)$ and negative $W^-(i,c;j)$ synaptic weights. The X represents the inputs vector to the network denoted as:

$$X = \{X_1, ..., X_n\} \text{ and } X_k = \{X_k(i,1), ..., X_k(i,C\}),$$

Being $X_k(i,c)$ the c-th class ($\forall c = 1, C$ and C=number of classes) of neuron i ($\forall i = 1, n$ and n=number of neurons) for pattern k, where $X_k(i,c) = \{\Lambda_k(i,c), \lambda_k(i)\}$. Y represents the vector of the desired outputs of the network, denoted as:

$$Y = \{Y_1, ..., Y_m\}, \text{ for } Y_k = \{Y_k(i,1), ..., Y_k(i,C)\},$$

in our approach, $Y_k(i,c)$ indicate LSBs intensities of each neuron, for example, Y(3) = (0.2, 0.8, 0.2) refers to the desired intensities output of the neuron 3 in the first, second and third class (0.2, 0.8 and 0.2, respectively). This LSB intensity is an approximation of the LSB value of each pixel in the watermark to be used by the MCRNN (see section 3.1.3). The initialization of this set of input and output pairs in the network is:

- The probability of each neuron i fires signal outside the network will be d(i, c) = 0.
- $\Lambda(i,c) = L_{ic}$ and $\lambda(i) = 0$, where L_{ic} is a constant for the class c of the neuron i, which is defined in the following section.
- Since $X_k(i,c) = \{\Lambda(i,c), \lambda(i)\}$, in our case $X_k(i,c) = (L_{ic}, 0)$.

3.1.2 Excitation function, exogenous signals and emission rate

This section defines the excitation probability for our model, which is fundamental in the training, detection and embedding processes. The probability of excitation of each neuron is given by [1]:

$$q(i,c) = \frac{\alpha(c)q(c) + L_{ic}}{r(i,c) + \beta(c)q(c)} \tag{9}$$

for $0 < q(i,c) < 1$, such that,

$$L_{ic} = \beta(c)q(c)(q(c) - 1) \tag{10}$$

$$\alpha(c) = \sum_{i,c;j,z} w^+(i,c;j,z) \tag{11}$$

$$\beta(c) = \sum_{i,c;j} w^-(i,c;j) \tag{12}$$

where q(c) is the average intensity levels of LSB present in the pattern of the watermark, which is explained below; L_{ic} is a constant for the class c from a neuron i which represents the arrivals of the exogenous signals to the neuron (in this work, L_{ic} is chosen according to the average intensities of the neuron classes, assimilating a learning behavior as that used for the recognition of colored images used in [1]), $\alpha(c)$ represents the sum of all positive synaptic weights of the network, and $\beta(c)$ represents the sum of all negative weights in the network. The emission rates r(i,

c) is defined as:

$$r(i,c) = \sum_{u=1}^{n}\sum_{p=1}^{C} w_k^+(i,c;u,p) + w_k^-(i,c;u) \tag{13}$$

The so-called probability of excitation of each neuron (equation 9) is the output of the MCRNN, which by updating will be trained to reach values very close to the desired outputs $Y_k(i,c)$.

3.1.3 Determination of the LSB intensity and average intensity q(c)

Since the LSBs values are always binary, we must map them to real values in order to be used on the equations of the section 3.1.2. We use the next rule: If we have a LSB binary value 0 then the intensity of this LSB will be 0.2 (20%), if the LSB binary value is 1 then the intensity will be 0.8 (80%). Obviously, these 2 values could be others, such as 0.1 and 0.9, or 0.4 and 0.6, but never the numbers 0 and 1. We use the first one in this work (0.2 and 0.8). Now we can compute the parameter q(c) as the average of all intensities belonging to the pattern of the watermark, noting that there are three classes per neuron, and each class contains the LSB of a channel of this pixel pattern.

3.1.4 The training mechanism

The rule for updating the synaptic weights is the classical gradient descent rule, which minimizes the minimum square error [8]:

$$E_k = 1/2\sum_{i=1}^{n}\sum_{c=1}^{C}(Y_k(i,c) - q_k(i,c))^2 \tag{14}$$

The rule for updating both positive and negative weights for each pattern k, is the defined in [1] as follows:

$$w_k^+(i,c;j,z) = w_{k-1}^+(i,c;j,z) - \mu\sum_{i=1}^{n}\sum_{c=1}^{C}(Y_k(i,c) - q_k(i,c))\frac{\partial q(i,c)}{\partial w^+(i,c;j,z)} \tag{15}$$

$$w_k^-(i,c;j) = w_{k-1}^-(i,c;j) - \mu\sum_{i=1}^{n}\sum_{c=1}^{C}(Y_k(i,c) - q_k(i,c))\frac{\partial q(i,c)}{\partial w^-(i,c;j)} \tag{16}$$

where.

$$\frac{\partial q(i,c)}{\partial w^+(i,c;j,z)} = \frac{q(i,c)w^-(i,c;j) + q(i,c)^2 w^-(i,c;j)}{(w^+(i,c;j,z) + w^-(i,c;j) + q(i,c)w^-(i,c;j))^2} \tag{17}$$

$$\frac{\partial q(i,c)}{\partial w^-(i,c;j)} = \frac{-(1 + q(i,c))}{(w^+(i,c;j,z) + w^-(i,c;j) + q(i,c)w^-(i,c;j))^2} \tag{18}$$

And $\mu > 0$ is a constant called ”learning rate”, which is between $0 < \mu < 0.9$. We update the weight until the minimum square error, from now MSE, reaches a preset value ϵ_o.

In general, the training process is acceptable when the values of q(i, c) are very close to Y(i, c). The importance of this training is that these values should not diverge much from the desired output values because the network will use a threshold function to decide what bit embed and detect in the embedding and detecting processes, respectively. In the digital watermarking theory is defined a key that is used for both embedding and detecting processes [3], in this work we consider the synaptic weights trained as this key (the training process encrypts the key in the MCRNN weights). The macro-algorithm for the training process is defined as:

```
Algorithm 1 Training Process
1. Initialize positive and negative synaptic weights with random values between 0 and 1.
2. Repeat for each class of each neuron:
    2.1. Configure input/output pairs ($X_k$, $Y_k$)
    2.2. Calculate the excitation function (Eq. 9)
    2.3. Calculate the MSE (Eq. 14)
    2.4. Update the positive and negative synaptic weights according to Eqs.15 and 16
3. Until the MSE is lower than a predefined threshold $\epsilon_o$
```

3.2 Neural-LSB Embedding process

The embedding process consist on that the watermark pattern is embedded in each pixel of the original image, using the version of the LSB based on the MCRNN model proposed in this work. For that, we use the MCRNN trained and a function called "threshold function". The "threshold function," is a crucial part in the process of embedding and detection of watermark, since it decides which LSB is associated with each excitation probability q(i, c). The definition of this threshold function is as follows:

$$(19) \qquad F_v(q(i,c)) = \begin{cases} 1 \text{ if } q(i,c) \geq T & \qquad 0 < T < 1 \\ 0 \text{ if } q(i,c) < 1 - T & \end{cases}$$

where T is the threshold. The value of the threshold T is between the 2 values of the LSB intensity (an equidistant position) to guarantee the same probability of chosen of the two possible intensity values. To calculate $F_v(q(i,c))$ we consider the MCRNN trained.

The embedding of the watermark using the MCRNN consists on modifies the LSB values in the original image according to the LSB values of the watermark pattern in MCRNN. This pattern is represented in the embedding process by the previously MCRNN trained (its synaptic weights are the key). In general, the embedding process uses an embedding function $F_i(F_v(q(i,c)), LSB)$, which depend on the values returned by the function threshold $F_v(q(i,c))$ and the LSB value of the original image, as follows:

$$(20) \qquad F_i(F_v(q(i,c)), LSB) = \begin{cases} P(p,c) + 1 \text{ if LSB=0 and } F_v(q(i,c)) = 1 \\ P(p,c) - 1 \text{ if LSB=1 and } F_v(q(i,c)) = 0 \end{cases}$$

where P(p, c) represents the pixel p of the original image in the channel c. Like there are three channels per pixel (RGB) and each channel is associated with a class c, the channel c of the pixel P(p, c) will be read completely (this is, the full 8 bits), so that, depending on the value threshold function $F_v(q(i, c))$ and the LSB on the original image, we embed the pattern of the watermark by adding or subtracting 1 to P(p, c). In this way we embed the watermark pattern in it.

For performance reasons (complexity of the learning procedure to encrypted the watermark), the watermark is small. The original image can be of any size, for this reason in our approach the watermark pattern is embedded in the original image often following a fit process as shown in figure 1. For example, if we have a original image of 8x8 pixel and a watermark image of 4x4 pixel, the watermark will be embedded as often as necessary to fill the original image (in the example 4 times). In general, we use original images with dimensions equal in width and height, these dimensions are multiples of 4, so that patterns can be embedded throughout the entire original image. Obviously, the size of the watermark must always be less than the original image size.

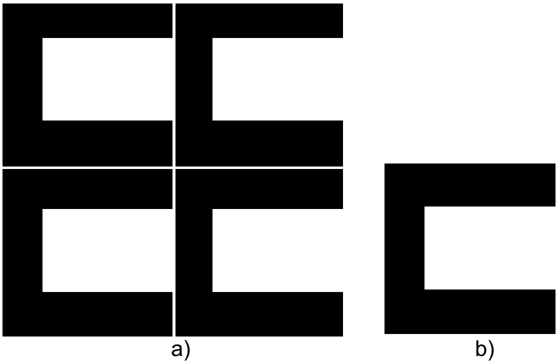


Fig. 1. a) Watermarks of 4x4 pixels embedded in an original image of 8x8 pixels and b) Watermark of 4x4 pixels

The next equation determines the total number of watermarks embedded in any dimension of an original image:

$$M_t = \left(\frac{S_t}{F_t}\right) B_t \qquad \text{for } F_t > 0 \tag{21}$$

where F_t is the total number of rows that has the original image, S_t is the total number of segments embedded in the original image and B_t is the total number of blocks that has the original image. The macro algorithm for the embedding process becomes:

Algorithm 2 *Embedding process*

1. Calculate the excitation probabilities q(i,c) of a MCRNN using the previously weights trained (the key)

2. Change the LSBs of the original image using the embedding function $F_i(F_v(q(i, c)), LSB)$ (see Eq. 20)

3.3 Neural-LSB Detecting Process

In our approach, the fundamental goal of the detecting process is to determine if all the watermark patterns previously embedded in the original image have not been changed. For that, we need to compare all embedded patterns with the pattern of the original watermark by using the MCRNN. We start with the computation of the probabilities of excitation of each neuron q(i,c) using the weights trained (key), to obtain the pattern of the original watermark. After calculated the probability of excitation q(i,c), we use the threshold function $F_v(q(i,c))$. These steps at this point are the same as in the embedding process. The next step is to compare the patterns embedded with the trained pattern. This comparison is performed by taking the LSB for each channel of each pixel of the carrier image and the value of the threshold function $F_v(q(i,c))$ related. This process is defined by a detection function $F_d(F_v(q(i,c)), LSB)$ as follows:

$$F_d(F_v(q(i,c)), LSB) = \begin{cases} \text{Intact Seal} & \text{if } F_v(q(i,c)) = LSB \\ \text{Broken Seal} & \text{if } F_v(q(i,c)) \neq LSB \end{cases} \tag{22}$$

The carrier image is declared intact if the condition $F_v(q(i,c)) = LSB$ is satisfied to all channels of every pixel of it. Otherwise (if we find a discrepancy in only one channel of any pixel, fulfilling that $F_v(q(i,c)) \neq LSB$ of the carrier image), the carrier image will be declared manipulated because the seal is broken. The macro algorithm for the detecting process is defined as:

```
Algorithm 3 Detecting process
1. Calculate the excitation probabilities q(i,c) using the previously trained weights (the key)
2. Detect the watermark rupture using the detection function $F_d(F_v(q(i,c)), LSB)$ over the carrier image (see Eq. 22)
```

4 Experiments

In general, the experiments will be performed by setting a threshold T with a value of 0.5, assuming an intensity of 0.2 and 0.8 for LSBs values of 0 and 1 of the watermark pattern, respectively. In our system the reading of the channels starting with the blue, then green and finally the red, and pixels are counted from left to right, starting from the bottom of the image.

The visual quality criteria to determine the effect of our approach over the carrier images, to be evaluated after embedding the watermarks in the original image, are the equations 23 and 24. This is necessary because the embedding process modifies the original images, affecting the visual quality of them. These criteria have been used by the scientific community to determine the distortion quality of the embedding approach of the watermarking techniques [3, 4, 7]. The formulas for these measures of distortion are:

- NG calculates the noise generated after embedding the watermark pattern in the original image. NG is defined as:

(23)
$$NG = \frac{\sum_{f,c}(Po(f,c) - Pm(f,c))^2}{3 * M * N}$$

where Po is the original image pixel, and Pm is the same pixel in the modified image after the process of embedding, M and N are the number of horizontal and vertical pixels of the original image, respectively, that are multiplied by 3 because we are working with the three RGB channels.

- PSNR determines the relationship between the maximum value that can have a pixel in an image and the noise generated by the process of embedding that affects the fidelity of its representation. Its definition is as follows:

(24)
$$PSNR = 10\log_{10}\left(\frac{MAXI^2}{NG}\right)$$

where MAXI represents the maximum value that a pixel can take (in this case 255). The return value of the PSNR is an indicator of the distortion caused by the embedding process. A low value means the changes undergone by the carrier image will be visible to the human eye or the HVS. The typical values of PSNR are between 30 and 50 decibels (dB), and a value less than 30 dB is an indicator of the need to adjust the embedding process, since the distortion of the carrier image will be evident. A PSNR value greater than 30 dB is considered acceptable [3, 4, 7, 12, 14].

4.1 Experiment 1: Analysis of the quality criteria of our approach

In this experiment the size of the original image is 96x96 pixels and the pattern image of the watermark is 8x8 pixels. On the other hand, there are three different images of watermark (see figure 3). The original image is the logo of the University of Los Andes (see figure 2).

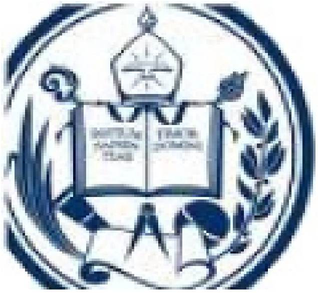


Fig. 2. Original image of 96x96 pixels.

4.1.1 Training Results

The following table shows the results of the training for different watermark, using a learning rate $\mu = 0.001$ for each case:

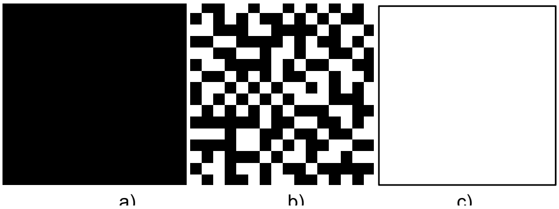


Fig. 3. Watermarks used. The watermark *b* is a random pattern

Parameter	Watermark a	Watermark b	Watermark c
Number of iterations	19	7	4
MSE	18,25%	24,22 %	10,03%
ϵ_o	20%	25%	15%
Neurons	9216	9216	9216

Table 1 Training results of the watermark patterns a, b and c in experiment 1.

Obviously, when training patterns are totally black or totally white, the results of the excitation probabilities tend to only one intensity of LSB. Moreover, the errors obtained in both training of a and c watermarks, with MSE 18.25% and 10.03%, respectively, are acceptable because of its proximity to the desired values, Y(i, c), are very close. However, these single-intensity patterns are highly dangerous to embed them as a watermark, because they lend themselves to possible manipulations that cannot be detected.

4.1.2 Embedding watermark b (figure 3.b) in the original image of 96x96 pixels

The following results are seen after the embedding process of the watermark *b* in the original image:

Embedded patterns= 144
PSNR = 51.5686 dB
Generated noise= 0.453

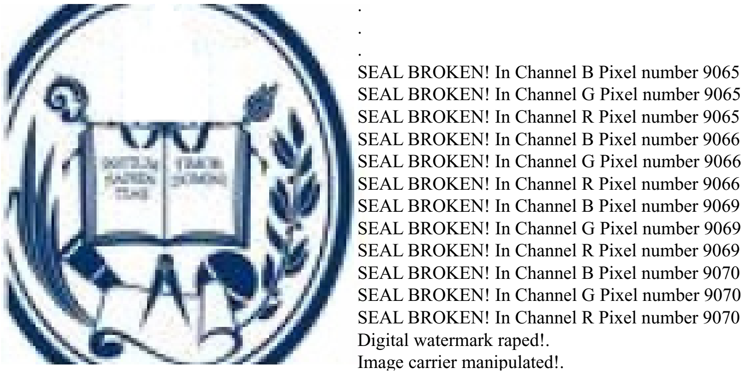
Fig. 4. Image carrier of 96x96 pixels.

144 watermarks patterns of 8x8 pixels were embedded in the original image of 96x96 pixels, with a PSNR of 51.5686 dB, which is a very good PSNR. In the LSB technique used in our work, the maximum difference that can exist between the value of a channel before and after embedding is only 1, that is, if we have a channel with a color of 244 for example (remembering that the color range is from 0 to 255), the maximum change in the color we can have in this channel would be 1 (so 244 in binary is 11110100, and changing the LSB of this channel from 0 to 1 being 11110101 in binary, the value of this channel in decimal will be 245). Thus, we see that the maximum possible noise in any image using the LSB technique is 1, because the assumption that an image of 96x96 all the channels were changed then the numerator of Eq. 22 would give 27,684, and this value is the number of

channels present in the image, which is defined as the denominator of Equation 22. If we calculate the PSNR with maximum possible noise, the maximum possible PSNR is 48.13 dB, for this reason is not expected a lower than 48.13 dB over any image using our approach. We can conclude that using our technique, it does not influence significantly in the distortion of the images embedded.

4.1.3 Detecting the watermarks of 8x8 pixels in the image carrier of 96x96 pixels

Now we proceed to manipulate the carrier image with the patterns previously embedded of the watermark b, deleting the crown at the top of the book (see figure 5). Our system returns the next messages:



```
.
.
.
SEAL BROKEN! In Channel B Pixel number 9065
SEAL BROKEN! In Channel G Pixel number 9065
SEAL BROKEN! In Channel R Pixel number 9065
SEAL BROKEN! In Channel B Pixel number 9066
SEAL BROKEN! In Channel G Pixel number 9066
SEAL BROKEN! In Channel R Pixel number 9066
SEAL BROKEN! In Channel B Pixel number 9069
SEAL BROKEN! In Channel G Pixel number 9069
SEAL BROKEN! In Channel R Pixel number 9069
SEAL BROKEN! In Channel B Pixel number 9070
SEAL BROKEN! In Channel G Pixel number 9070
SEAL BROKEN! In Channel R Pixel number 9070
Digital watermark raped!.
Image carrier manipulated!.
```

Fig. 5. Carrier image manipulated of 96x96 pixels.

We can see that our system detected the manipulation successfully. Additionally, we can know where exactly has been manipulated the image (pixels which have been modified).

4.2 Experiment 2: Lenna image

The original image is Lenna (see figure 6.a) of 256x256 pixels, using like watermark another woman (see figure 6.b) of 16x16 pixels.

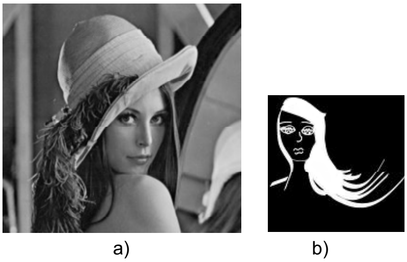
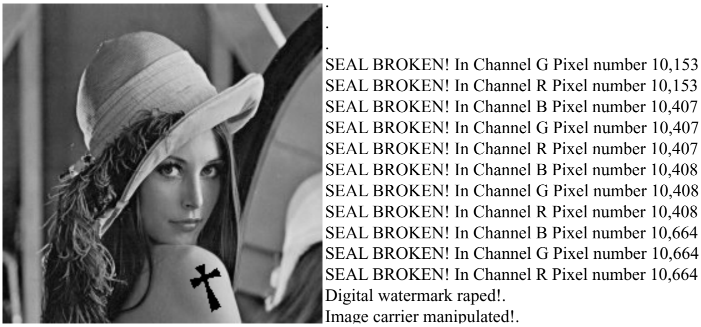


Fig. 6. a) Original image of Lenna of 256x256 pixels; b) Watermark of 16x16 pixels.

With a learning rate $\mu = 0.001$, $\epsilon_0 = 15\%$, a number of iterations=4, for 65536 Neurons, was reached a training with a 13.82% error (MSE). Additionally, 256 watermarks of 16x16 pixels each one are embedded. The values obtained are: PSNR = 51.2441 dB, Generated noise= 0.488. The value of PSNR is very similar. Thus, we can say then, that in our approach, the distortion obtained after the embedding process always tends to a PSNR of 51.14 dB, with an average noise of 0.5.

4.2.1 Detecting the watermarks of 16x16 pixels in the image of 256x256 pixels

Having embedded the watermark of figure 6.b and manipulating the image of Lenna drawing a tattoo at the right arm (see figure 7), the system returns the following:



```
.
.
.
SEAL BROKEN! In Channel G Pixel number 10,153
SEAL BROKEN! In Channel R Pixel number 10,153
SEAL BROKEN! In Channel B Pixel number 10,407
SEAL BROKEN! In Channel G Pixel number 10,407
SEAL BROKEN! In Channel R Pixel number 10,407
SEAL BROKEN! In Channel B Pixel number 10,408
SEAL BROKEN! In Channel G Pixel number 10,408
SEAL BROKEN! In Channel R Pixel number 10,408
SEAL BROKEN! In Channel B Pixel number 10,664
SEAL BROKEN! In Channel G Pixel number 10,664
SEAL BROKEN! In Channel R Pixel number 10,664
Digital watermark raped!.
Image carrier manipulated!.
```

Fig. 7. Image of Lenna manipulated.

Our system has detected the manipulation satisfactorily.

4.3 Analysis of Classical Attacks

Below is a table with a list of attacks classical over images. These attacks have been performed (simulated) over an original image of Lenna of 256x256 pixels over which we have used different watermarking approaches. In the case of our approach, we use the random watermark pattern of the figure 5. This table shows if each technique detects the attack or not.

Attacks vs. works	Our approach	[5]	[7]	[13]	[11]
JPG compression	Yes	Yes	Yes	Yes	No
Horizontal Flip	Yes	No	Yes	-	No
Rotation	Yes	No	Yes	Yes	No
Cropping	Yes	Yes	No	Yes	Yes
Scaling	Yes	Yes	No	Yes	Yes
Brightness and contrast correction	Yes	-	No	No	Yes
Gaussian Noise	Yes	No	Yes	Yes	Yes

Table 2. Comparison of the results of the attacks over the carrier images in different works

All the attacks over the carrier image were detected by our approach. For the rest of works, only some of them are detected (in some case, they do not detect some attacks). In the case of the SVD approach [5], it has difficult to detect a violation when the image is moved or if we introduce some noise because this matrix approach cannot detect some of the modification. The behavior of DCT and DWT approaches [11] is similar to the previous one due to the transformation gives problems to take in account these types of modifications. The watermarking technique based on artificial immune recognition system [7] has problem to detect modifications on the images because this technique require a large time to create the cell of violations. Finally, the neural networks of [13] has problem when we modify the image in terms of Brightness and contrast correction (the learning procedure is not robust for this modification).

Our approach can detect different types of attacks to an image without a perceptible cost. The main cost is the execution cost at the beginning to encrypt the watermark (learning phase); the embedding and detection processes are very fast.

5 Conclusion

The coupling of the LSB technique in the MCRNN model to detect unauthorized manipulations in images was carried out, obtaining satisfactory results. The comparison of our work with previous one is very encouraging (with detect a lot of attacks, our approach does not generate a large distortion to the images, etc.). With regard to the sensitivity of the parameters of our approach, we have observed that the behavior of the learning rate does not affect the MSE obtained, since for large and small rates was possible to obtain successful trainings with approximately equal MSEs. For this reason it was fixed $\mu = 0.001$. On the other hand, the ?o error threshold was set to 30%, as the maximum tolerable error in the experiments discussed previously. Finally, the threshold T was set to 0.5, and LSB intensities of 0.2 and 0.8 for bits 0 and 1, respectively.

The times of the processes involved (training, embedding and detection), were not analyzed in this work, since their values in milliseconds were too small. For example, in several of the experiments the longest training execution time was 1235 milliseconds to 5 iterations, for a watermark of 16x16 pixels. The embedding and detection processes had the same behavior.

As future work we propose the development of new prototypes that can connect the model presented in this work with other digital objects such as text, audio and video. This increases the possibilities of using this model in the area of computer security. We have tests our system for BMP images, we need extend it to images in other formats (such as JPG, GIF and PNG).

References

[1] Aguilar, J. *A Color Pattern Recognition Problem Based on the Multiple Classes Random Neural Network Model.* Neurocomputing, **61** (2004). 71-83.

[2] Anaculate M. and Aloysius G. *Novel and Robust Approach for Digital Watermarking Using Lifting Wavelet Transform*, American Journal of Scientific Research, American Journal of Scientific Research, **13** (2011) 118-130.

[3] Arnold, M., Schmuker, M. and Wolthusen, S. "Techniques and Applications of Digital Watermarking and Content Protection". Boston: Artech House, 2003.

[4] Caldelli R., Filippini F. and Becarelli R. *Reversible Watermarking Techniques: An Overview and a Classification*, EURASIP Journal on Information Security, **2010** (2010) 19 pages.

[5] Chandra B. and Srinivas S. *Robust Multiple Image Watermarking Scheme using Discrete Cosine Transform with Multiple Descriptions*, International Journal of Computer Theory and Engineering, **1** (2009). 527-533.

[6] Cox, I., Miller, M., Bloom, J., Fridrich, J., and Kalker, T. "Digital Watermarking and Steganography". 2nd ed., Boston: Morgan Kaufmann Publishers, (2002).

[7] Findik O., Babaoglua I. and lker E. *A color image watermarking scheme based on artificial immune recognition system*, Expert Systems with Applications, **38** (2011).1942-1946.

[8] Fourneau, M., Gelenbe, E., and Suros, R. *G-networks with multiple classes of negative and positive customers.* Theor. Comput. Sci. **155** (1996) 141-156.

[9] Gelenbe, E. *Random neural networks with positive and negative signals and product form solution.* Neural Comput. **1** (1989) 502-511.

[10] Gelenbe, E. *Theory of the random neural network model*, in: E. Gelenbe (Ed.), Neural Networks: Advances and Applications, North-Holland, Pays-Bas, 1991.

[11] Jianshengn M., Sukang L. and Xiaomei T. *A Digital Watermarking Algorithm Based On DCT and DWT*, Proceedings of the 2009 International Symposium on Web Information Systems and Applications, (2009) 104-107.

[12] Majumder S., Shankar T., Mankar V., and Sarkar S. *SVD and Neural Network Based Watermarking Scheme*, Information Processing and Management Communications in Computer and Information Science, **70** (2010) 1-5.

[13] Najafi H. *A neural network approach to digital data hiding based on the perceptual masking model of the human vision system*, International Journal of Intelligent Computing and Cybernetics, **3** (2010).391-409.

[14] Rini T. *Review of Robust Video Watermarking Techniques*, International Journal of Computer Applications: Special Issue on "Computational Science-New Dimensions & Perspectives" (2011) 90-95