An Attack Proof Intelligent Digital Watermarking Based on Safe Region of Fast Fourier Transform

R.F. Olanrewaju[1], Othman.O. Khalifa, Aisha-Hassan Abdulla and A.A Aburas
Department of Electrical and Computer Engineering,
Faculty of Engineering
International Islamic University Malaysia (IIUM)
P O Box 10, 50728 Kuala Lumpur, Malaysia.
[1]frashidah@yahoo.com

Abstract— **Recent advancement in digital medium has created a need for secure transfer and transaction over the internet. The fact is that the current digital distribution and storage technologies are great threat to multimedia industries where unlimited number of perfect copies can be produce illegally. In this paper we discuss digital watermarking as a means of hiding owner's copyright message in images. At the moment, the most critical issue faced by the watermarking system is determining the best place to hide watermark data. We propose a method, an attack proof intelligent system, in which Artificial Neural network is use to locate the Safe Region in the host image and the watermark is embedded based on the located Safe Region in Fast Fourier Transform domain. Experiment on a large set of natural images shows the robustness of the new scheme. The implementation results have shown that this watermarking algorithm has high level of imperceptibility and the watermark bit were all recovered correctly.**

Keywords: **(Artificial Neural Network, Back Propagation algorithm, Fast Fourier Transform, FFT, Image watermarking, Safe Region,)**

I. INTRODUCTION

With the wide spread, complex use, and transfer of digital media, secure media transfer has been a concern to all the multimedia industries. This concern is appropriately addressed by digital watermark. Digital watermarking is a novel approach that involves embedding of digital mark into a multimedia object (cover work) such that it is robust, secure and imperceptible to the human observer, but can be detected algorithmically [1]. Due to digital watermark crucial features such as; imperceptibility, inseparability of the content from the watermark, and it's intrinsic ability to undergo same transformation as experienced by the cover work, has made it superior and preferable over other traditional methods of protecting data integrity, authentication of information resources, ownership assertion, confidentiality, copy protection, data monitoring and tracking. This preference has been proven experimentally [2] to provide improved security. Digital watermark has two generic building blocks. The watermark embedding block also known as encoder with a respective watermark recovery blocks also known as decoder. The embedding block inserts the watermark information in the data while the recovery blocks extract/decodes the watermarked information. Figure 1 depicts the digital watermarking system showing the embedder and extractor.

Figure 1. A watermarking system with embedder and extractor.

Recent works have taken advantage of artificial intelligence in Neural Networks to improve the security and design a robust watermarking system [3]-[6]. Owing to the inherent characteristic of Neural Network like learning and adaptive capabilities, pattern mapping and classification and ability to generalize, not only to reproduce previously seen data, but also provide correct predictions in similar situations gives the trained networks ability to recover the watermark from the watermarked data. Examples of application of ANN in watermark include capacity estimator, error rate prediction, embedding and recovery of mark, detection of tempering etc.

II. RELATED THEORIES

A. Artificial Neural networks and watermarking

Artificial Neural Network (ANN) as emerged as a powerful tool for computational model based on biological neural networks [7], in other words, is an emulation of biological neural system. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. It starts by transmitting input signals through the connection; each connection has an associated weight that improves the transmitted signal; each neuron transforms the received signals through an activation function to which in turns determine the output signal. In most cases ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. ANN can learn from the data and generalize things learned. They extract the essential characteristics from the numerical data as opposed to memorizing all of it. This offers a convenient way to reduce the amount of data as well as to form an implicit model without having to form a traditional, physical

model of the underlying phenomenon. While there are numerous different artificial neural network architectures such as Single layer feed forward, Multilayer feed forward, fully recurrent network, competitive network, Jordan Network and Simple recurrent Network etc have been studied by researchers. The most successful applications in data mining, classification, recognition etc of neural networks have been multilayer feed forward networks [8]. Figure 2 shows a typical architecture of multilayer feed forward neural network.

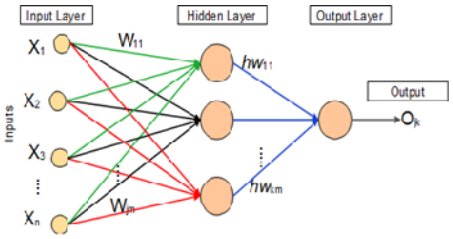

Figure 2. General Architecture of Feed forward NN

Multilayered perception NN models in watermarking and steganography are effectively employed at the Embedding, detection, extraction and location of Secure Region (SR) within the host image. For instance very recent, Olanrewaju *et al* [1] have shown that NN can be use to locate a Secure Region (SR) within the host image. SR is an identified region in the host media in which when watermark is hidden therein, it will not be destroyed nor degraded. It was found that watermark detection error rate help embed more watermark messages while keeping an acceptable detection error rate, and it is useful for the design of the general algorithms of watermarking and detection. According to the experimental results obtained by Zhang et al [9], the detection error rate of watermark is mainly influenced by the watermark average energy and the watermarking capacity. The error rate rises with the increase of watermarking capacity. When the channel coding is used, the watermarking error rate drops with the decrease of the payload capacity of watermarking. Naoe and Takefuji [10] proposed a frequency based transform watermarking using NN on YCbCr domain to detect a hidden bit codes from the content. A conditioned neural network is used as a classifier to recognize a hidden bit pattern from the content which embedder associated to the target content. They found that the method does not damage the target content. Though, the extraction keys must be shared among embedder and extractor in order to extract a proper hidden bit codes from the target content.

B. Back Propagation Algorithm (BP)

BP is one of the algorithms which have hugely contributed to neural network fame. The principal advantages of back propagation are simplicity and reasonable speed (though there are several modifications which can make it work faster, [11]. Back-propagation is well suited to pattern recognition, classification and detection problems. A BP network learns by example, that is, we must provide a learning set that consists of some input examples and the known-correct output for each case. So, we use these input-output examples to show the network what type of behavior is expected, and the BP algorithm allows the network to adapt. The BP learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random). This output is compared to the known-good output, and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point we say that the network has learned the problem ”well enough” the network will never exactly learn the ideal function, but rather it will asymptotically approach the ideal function. A typical error plot for BP is as shown in Fig. 3

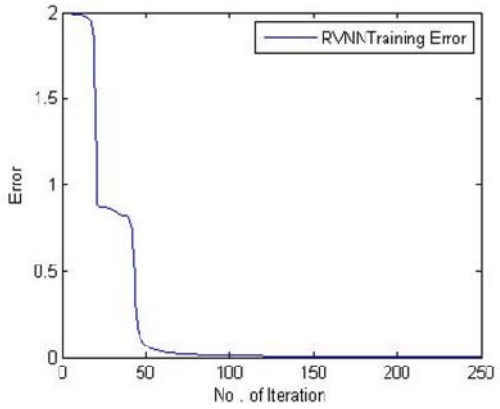

Fig. 3. Typical error plots for BP Algorithm

C. Backpropagation Weight Update Formulation

The block diagram of Real Valued Neural Network (RVNN) back propagation shown in Fig. 4 for a Real Valued Data (RVD) weights update equation is developed as follow:

$$e(n) = T(n) - y(n) \quad (1)$$

Where $T(n)$ is the target RVD and $y(n)$ is the output of RVNN. The objective is to find the set of parameters that minimize the sum of the squared of the error function, that is:

$$E(n) = \frac{1}{2}\sum_{n=1}^{l} e(n)^2 \quad (2)$$

l is the total number of neuron in the output layer. The network weight update is given by;

$$w(n+1) = w(n) + \partial w(n+1) \quad (3)$$

Where

$$\Delta w = -\mu \nabla_{\mathrm{w}} E_n \tag{4}$$

and μ is the learning rate, $\nabla_{\mathrm{w}} E$ is the gradient of the cost function. Therefore, (4) can be re-written as a partial derivative given by:

$$\nabla_{\mathrm{w}} E_n = \frac{\partial E_n}{\partial w_n} \tag{5}$$

In order to use the chain rule to find the gradient of the error function *E* with respect to *w*, the interdependency of the variables need to be taken into consideration, the partial derivatives can be written as:

$$\frac{\partial E_n}{\partial w_n} = \frac{\partial E_n}{\partial_y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial w_n} \tag{6}$$

Let

$$\partial(n) = \frac{\partial E_n}{\partial_y} \tag{7}$$

Then to solve equation (5) involves evaluating the values of all the partial derivatives contained in equations (6). Detail of the derivates id presented by the authors in [12]

Figure 4. RVNN back propagation scheme.

III. MATERIALS AND METHODS

A. *Fast Fourier Transform (FFT) Safe area features for image watermarking*

Given an image *I(x,y)* of size *MXN*, for *x* = 0, 1…*M-1* for *y* = 0, 1…*N*-1. The 2-D Discrete Fourier Transform (DFT) of *I* is represented by *F*(*u, v*)

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I[x,y] e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \tag{8}$$

Thus given *F*(*u v*), we can obtaining *I*(*x, y*) back by means of the Inverse 2Dimensional DFT

$$I(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})} \tag{9}$$

where *u,v* are frequency variables and *x, y* are spatial variables.

In this work, the FFT is applied on the host image so as to convert the spatial domain image to frequency domain for analysis and detection. Windowing is also applied to the frequency spectrum of the data, this is to prevent spurious peak from appearing in the spectrum. To increase the readability of the spectrum, zero padding was applied after windowing for incomplete window frame.

FFT Block decomposition and transformation of the host image is the first step in FFT process. The host image is decomposed into non-overlapping 8 X 8 blocks. An *N* point FFT transformation of each selected and decomposed block is implemented independently in this stage as the input vector space . For the host image *I* with *M*x*N* size, there should be *N*2 FFT point such as 16, 64, 128, 256 etc. This is because the FFT point must not be smaller than the data length. FFTshift of each block was taken to determine the DC component. Once the data are sorted, the midpoint is easily located. This procedure help in partitioning the data into left hand side LHS and right hand side RHS based on the target data. For data on RHS of the number line, it is considered positive with a target data 0 while LHS is negative and it target is 1. Further categorization is done on both RHS and LHS based on frequency component in to low, mid and high frequency.

B. The Neuron Model

The neuron as shown in Figure 5 is divided into two part: the summer and the activation function part. It begins by summing up the weighted input in other to obtain the threshold. The resultant sum is fed into the activation function which maps weighted sum to the real value output. A model of the neuron use in this work is shown in Figure 5.

Figure 5. Model neuron

C. Activation function

The neuron uses sigmoid activation function. As soon as the result from summer is fed into the activation function, the activation function is triggered and makes the output bounded. The activation function is defined as:

$$A = \frac{1}{1+exp^{-(net)}} \tag{10}$$

Figure 6.FFNN activation function

Various activation functions are shown in Figure 6. For this work, sigmoid activation function is used.

B. Training and learnng convergency

At the beginning of the training, weights and bias were initialized with small random values. After several experimentation, the optimum architecture (shown in Figure 7) nodes and learning rate were found to be 11:5:1 and 0.49 respectively.

Figure 7. Neurons topology

Learning convergence

Generally when any neural network is trained with inputs, the error on training dataset decreases gradually with the epochs or goal set. The RVNN stops training when the goal is archived or reached the maximum number of epoch specified, which ever condition is meet first. Equation 11 was used as criteria to stop the training when maximum epoch set is achieved (250epoch). Ones the condition in Equation 11 is satisfied, the algorithm stopped training and weight and bias are blocked. Figure 4 shows how the mean square error changed over the epochs.

$$E = \frac{1}{2j}\sum_{j=1}^{j}\sum_{i=1}^{n}(T_{ji} - y_{ji})^2 = 250 \qquad (11)$$

C. Watermark Embedding Method

1. Block selection: Consider a grayscale host image *I* with size *M* x *N* and a watermark *w* of size *Mw x Nw* binary image. The total block that can be selected from *I* is $\frac{M}{8}$ x $\frac{M}{8}$ size 8x 8. However, the size of the block to be selected for watermarking is determined by the total size of the watermark *w*. The selection can be sequential or random.
2. Transformation of the each selected block to obtain frequency components for the for the RVNN is explained in section IIIA.
3. Establishing a RVNN relationship between the input data and the target content, which is, carefully matching the target content with the input data. An 11x5x1 RBN was established for training purposes as shown in Figure 7. A three bits binary target content *T* is mapped to the input data. The mapping rule is determined by:

$$T = \{t_{(m,n)} | t_{(m,n)} \epsilon \{0,1\}\}$$

$t_{(m,n)}$is defined as follow:

$$t_{(m,n)} = \begin{cases} 1 mid \\ 0, if low or high \end{cases}$$

Where T is the target content and $t_{(m,n)}$ is the position of low, mid and high frequency.

4. Modification of the FFT coefficient.
5. Training is repeated and the relationship is adjusted between the target content and the corresponding output of RVNN model until the network learning threshold is satisfied and the convergent of network weights are archived.
6. Weights are saved for future extraction.

D. Watermark Extraction Method

The extraction processing is similar to the embedding process though in reverse order. However the IFFT is not required.

1. Block based transformation of the watermarked image using FFT. The watermarked image *I`* and the host image *I* are transformed independently.
2. The middle frequency coefficients of each block are located
3. The saved weights are used to extract the watermark.
4. Correlation between the original watermark and extracted watermark is calculated.

IV. RESULT AND DISCUSSION

The images use is shown in Figure 8, taking the FFT of the images gives both real and imaginary values. The real values were used for both training and testing. The multilayered RVNN is configured with one hidden layer and the hidden neurons in the hidden layer are 5, the

diagram of which is as shown in Fig. 7. We carried out a series of simulations to test the algorithm. The 512 x 512 grayscale image pepper shown in Fig 8a was used as the cover image and a 32 x 32 binary watermark image (IIUM logo) is shown in Fig. 8b. A RVNNBP learning rule discussed in section IIB was used for RVNN, the training epoch was set to be 250 and the learning rate was kept to be 0.49. Weights and bias were initialized with small random numbers. 20% of the original data was use for training the network. 0nce the network is converged, the next 40% were used as test data while the last 40% was also used for validation.

The performance of the algorithms is appraised by some objective performance measure such as: Image Fidelity Measure IFM for the extracted watermark. IFM is used as accuracy indicator for the retrieved watermark. Technically, IFM is a similarity measurement between two different signals. The value ranges between 0 - 1. When the result of two signals/images is 1 it means they are similar while 0 means dissimilar, that is, higher value signifies closeness.

IFM is defined as:

$$IFM = 1 - \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}|U\big((m,n) - V(m,n)\big)|^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}(U(m,n))^2}$$

The results of watermarking shown in Table 1

Since there is no physical embedding of any data into the cover image, there will be no visual quality degradation to the watermarked image as shown in figure 9. In fact, the watermarked image is indistinguishable from the original host image. Here, we only discuss the accuracy of mapping and watermark recovery.

.

The algorithm is reliable since it gave the highest percentages of correctly mapped watermark bit of 99%. That is all the watermark bits are correctly mapped to the host image position.

Figure 8. Pepper, the host image and IIUM logo as the watermark

Figure 9. watermarked image

From results obtained in figure 9 above, the watermarked image is highly imperceptible, that is the watermarked image and the cover images (8a) are alike without any visual degradation. This is because of the watermarking strategy. It can also be seen from the extractor block, clearly that the extracted watermark bits are all recovered without damage to the host image because the mapping regions were carefully selected.

V. CONCLUSION

This study suggests an efficient and distortion free digital watermarking algorithm using neural network and FFT. A mapping strategy is used in place of conventional embedding (+) structure, which has inherent advantage in terms of watermark recovery, distortion free of the host image requirements. In addition, exhaustive simulation results indicate that the proposed method was able to recover all the watermark bits and if watermark is tempered, it can locate each block and position that is tempered. Varying the epoch, topology of the network as well as testing the effect of various activation functions on the algorithm is enumerated as a future work problem.

ACKNOWLEDGMENT

This work was supported in part by International Islamic University Malaysia (IIUM) research endowment grant number: EDW B 0902-215

REFERENCES

[1] R.F Olanrewaju, A. A Aburas, O. O. Khalifa and A. Abdalla, "State-of-the-Art Application of Artificial Neural Network in Digital Watermarking and the Way Forward", Inter. Conf. on Computing and Informatics, 2009, pp. 233-237, June 2009.

[2] T. Schmidt, H. Rahnama and Sadeghian, A. A Review of Applications of Artificial Neural Networks in Cryptosystems, *World Automation Congress*, 1-6, 2008.

[3] Shih Y. F. & Wu, Y. T(2005). Robust Watermarking and Compression for Medical images based on Genetic Algorithms. . *Int. Journal of Information Sciences*, 175, 200–216

[4] Ting, G. C. W., Goi, B. M. & Heng, S. H. (2007). A Fragile Watermarking Scheme Protecting Originator's Rights for Multimedia Service. *Lecture Notes in Computer Science on Computational Science and Its Applications*, 4705/2007, 644-454.

[5] Tsai, H. H. (2007). Decision-Based Hybrid Image Watermarking in Wavelet Domain Using HVS and Neural Networks. *D. Liu et al. (Eds.): ISNN*, Part III, 4493, 904–913.

[6] Wen, X. B., Zhang, H., Xu, X. Q. & Quan, J. J. (2008). A New Watermarking Approach Based On Probabilistic Neural Network

In Wavelet Domain**.** *Soft Computing-A Fusion of Foundations, Methodologies and Applications***,** 13, 4, 355-360.

[7] Haykin, S. (2008). Neural Networks and Learning Machines**,** 3rd edition. Prentice Hall.

[8] A. K. Palit and D. Popovic, " Computational Intelligence in Time Series Forecasting", Springer. 2005.

[9] F. Zhang, X. Zhang and H. Zhang, " Digital Image Watermarking Capacity and Detection Error Rate", Pattern Recognition Letters, pp.1-10, 28, 2007.

[10] K. Naoe and Y. Takefuji, " Damageless Information Hiding using Neural Network on YCbCr Domain", Int. Journal of Computer Science and Network Security, pp 8 9 2008.

[11] H. Lari-Najafi, M. Nasiruddin and T. Samad, "Effect of initial weights on back-propagation and its variations", IEEE Inter. Conf. on Systems, Man and Cybernetics, 1989. , pp.218-219 vol.1, 14-17 Nov 1989.