DAMAGELESS DIGITAL WATERMARKING USING COMPLEX-VALUED ARTIFICIAL NEURAL NETWORK

Rashidah Funke Olanweraju[1], Abdurazzag Ali Aburas, Othman Omran Khalifa and Aisha-Hassan Hashim Abdalla

Faculty of Engineering
International Islamic University Malaysia

frashidah@yahoo.com[1]

ABSTRACT

Several high-ranking watermarking schemes using neural networks have been proposed in order to make the watermark stronger to resist attacks. However, the current system only deals with real value data. Once the data become complex, the current algorithms are not capable of handling complex data. In this paper, a distortion-free digital watermarking scheme based on Complex-Valued Neural Network (CVNN) in transform domain is proposed. Fast Fourier Transform (FFT) was used to obtain the complex number (real and imaginary part) of the host image. The complex values form the input data of the Complex Back-Propagation (CBP) algorithm. Because neural networks perform best on detection, classification, learning and adaption, these features are employed to simulate the Safe Region (SR) to embed the watermark, thus, watermark are appropriately mapped to the mid frequency of selected coefficients. The algorithm was appraised by Mean Squared Error MSE and Average Difference Indicator (ADI). Implementation results have shown that this watermarking algorithm has a high level of robustness and accuracy in recovery of the watermark.

Keywords: Digital Watermarking, Complex Back Propagation Algorithm, Complex-Valued Data (CVD), Complex-Valued Neural Network (CVNN), Fast Fourier Transform (FFT).

INTRODUCTION

Digital watermarking is an approach that involves the embedding of a digital mark into a multimedia object (cover work: image, audio, video text) such

that it is robust, secure and imperceptible to the human observer, but can be detected algorithmically. Due to the digital watermark's crucial features such as imperceptibility, inseparability of the content from the watermark, and its intrinsic ability to undergo the same transformation as experienced by the cover work, this has made it superior and preferable over other traditional methods of protecting data integrity, authentication of information resources and ownership assertion. This preference has been proven experimentally (Schmidt, Rahnuma & Sadeghian, 2008) to provide improved security. An additional cofounding factor of a watermarking system is the robustness of the algorithm against attacks. Robust watermarks are designed to survive common distortion and resist malicious attacks (Kuttera & Petitcolas, 2000). All applications presupposing security of the watermarking systems require this type of marks in order to survive any kind of alterations or intentional removal introduced by standard or malicious processing and attacks. Examples of robust watermarking can be found in the works of (Lu, Sun & Lu, 2009, Celik, Lemma, Katzenbeisser & Veen 2008; Shih & Wu, 2005). Something is missing Some watermarking schemes embedded with very low robustness are called fragile watermarks. Fragile watermarks can be destroyed with the slightest manipulations, as long as the document has been illegally manipulated. Such types of watermarks are used to check the integrity of objects and might be useful if digital images are used as evidence in litigation and forensic application. It is also used to verify the medical content of medical images due to its sensitivity (Zain & Fauzi, 2007). Applications that use fragile watermarks can be found in the work of (Ting, Goi & Heng, 2007). A watermarking scheme is mostly designed in view of its application, and there is no such watermarking scheme that can perform well under all hostile attacks (Khan, Tahir, Majid & Choi, 2008). For example, (Parthasarath & Kak, 2007) designed a DCT content-based watermarking scheme. In their method, texture, luminance, corner and the edge information in the host image were used to generate the mask of a watermarking scheme. It was found that their scheme was robust against JPEG compression, median filtering and contrast-sharpening filter. However, it was not robust against scaling, rotation and high noise levels. For these reasons, the development and the evaluation of watermarking algorithm is a challenging task.

Least Significant Bit (LSB) coding was among the first techniques used in watermarking (Van, Tirkel & Osborne, 1994), an today till LSB is a common and easy-to-use watermarking scheme. LSB coding is easily achieved by camouflaging the noises inherent in to digital-signal acquisition process (Gao & Gu, 2007). When LSB came into use, it was as if it couldnot be attacked. However, with advances in algorithm, various steganalysis surfaced (Ge, Gao & Wang, 2007, Luo, Liu & Liu, 2005) which found LSB to be vulnerable

to even the slightest image distortion. Researchers later improved LSB by introducing algorithms such as Patchwork, a statistical method of embedding (Bender, Gruhl, Moto & Lu, 1996), correlation-based techniques, feature points and chaotic maps (Peng & Jiang, 2009).

However, due to direct manipulation and linear addition of the watermark to the host media of the above techniques (LSB, Patchwork and chaotic map), are vulnerable to attacks. Such techniques are also known as spatial domain techniques. Spatial domain techniques are susceptible to attacks like compression, geometric distortation, image degradation as well as computational inefficient. To obtain better imperceptibility as well as robustness, the embedding of the watermark is done in transformed domain. Scientists exploited the benefits of transform domains like Discrete Cosine Transform (DCT) (Choi Seo, Yoo & Kim, 2008), Discrete Fourier Transform (DFT) (Sang and Alam 2008), Hadamard Transform (Abdallah et al., 2006), and Discrete Wavelet Transform (DWT) (Senthil & Bhaskaran, 2007) to build a robust watermarking algorithm. These current schemes are not totally robust against all attacks. However, when the principles of neurocomputing, and their usage in science and technology surfaced, (Ham & Kostanic, 2001), the use of a Neural Network (NN)-based watermarking scheme was successful. This is because a NN-based scheme performs well under specific sets of conceivable attacks and work well with Human Visual System (HVS).

ARTIFICIAL NEURAL NETWORK IN WATERMARKING, RELATED WORK

Recent works have taken advantage of artificial intelligence in Neural Network to design a robust watermarking system. Owing to the inherent characteristics of Neural Network like learning and adaptive capabilities, pattern mapping and classification as well as its ability to generalize, not only reproduce previously seen data, but also provide correct predictions in similar situations gives the trained networks the ability to recover the watermark from the watermarked data. Examples of applications of ANN in the watermarking system include embedding and recovery of mark, capacity estimation (Me et al., 2002, Fang & Zhang, 2005), error-rate prediction (Zhang & Zhang 2007), detection of tempering, Safe Region location (Olanrewaju et al., 2010).

Artificial Neural Network in Embedding and Recovery of Watermark

In ANN embedding, transform domains such as DCT, DWT, DFT etc., can be used to decompose the host media coeffcients first. Then a chosen NN can be used to train these information. Such training will establish the relationship

among the coefficients that will serve as an input to the layers of network. Finally, the watermark sequence is embedded into the host media adaptively by adjusting weights to produce corresponding target watermark media at the output layer. This watermark can be stored at the same time if the original media is necessary watermark detection (Bansal et al., 2005).

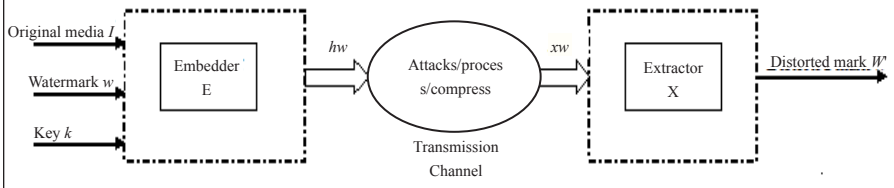

Figure 1. Block Diagram of Embedder and Extractor

A typical watermarking scheme is illustrated in Figure 1 showing the two main building blocks; anembedder/encoder with a respective extractor/decoder. In general, the embedding process can be understood as the combination of a watermark signal and the original host (image, video, text audio) media. The watermark embedder inserts a watermark into the cover signal and the watermark recovery block extracts/decodes or detects the presence of the watermark signal. To create a watermarked image *C(x, y)*, the pseudorandom pattern *W(x, y)* is multiplied by a small gain factor k and added to the host image or cover data *C(x, y)*, as shown in equation 1 illustrated in Figure 2.

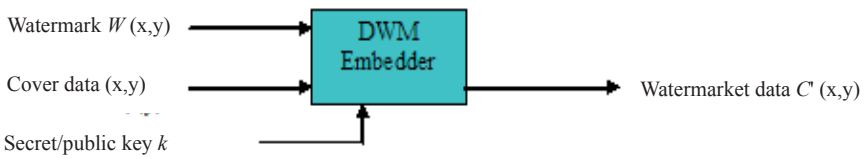

Figure 2. Digital Watermarking Embedding Scheme

$$C^{\wedge}{}'(x,y)=C(x,y)+k.W(x,y) \quad (1)$$

Li and Wang (2007), demonstrated the combination of the video-watermarking scheme based on RBF Neural Network and 3D-DWT. The algorithm established

a relationship among the coefficients of the discrete wavelet decomposition along the spatial and temporal axis using the RBF networks. It was applied during the embedding and extracting processes to limit pirate attacks. Chang (2005) proposed a Full-Counterpropagation Network (FCNN) to insert the watermark into the synapses of FCNN rather than the cover image. This method increased the robustness of the algorithm and reduced imperceptibility problems to a great extent. Similarly, Bansal and Bhadauria (2005) introduced a scheme based on Backpropagation Neural Network to train a given cover image in which the trained-network weights are hidden within the original cover image itself. This increased robustness and preserved the cover image. Majhi and Shalabia (2005), also showed how to again computation efficiency as well as memory requirement by embedding and detecting watermarks using a modified Functional Link Artificial Neural Network (FLANN) in Discrete Cosine Transform (DCT). The algorithm was trained using backpropagation algorithm to achieve minimum MSE. The results showed that the scheme reduced computational cost in the training stage and maintained a good performance of approximation. Tsai (2007) proposed a decision-processor-based watermarking in the wavelet domain using NN that incorporated the HVS model. This technique was used to extract the watermark and enhance imperceptibility of the watermarked media.

Artificial Neural Network as Capacity Estimator

Researchers like Fang and Zhang (2005) studied the bounds of embedding capacity in a blind watermarking algorithm based on the Hopfield neural network. They used basin attraction via Hamming distance to restrict the capacity of the watermark. Similarly Shi-Chun et al., (2002) modeled the Human Visual System (HVS) using the feed forward ANN-based image-adaptive method in order to decide the watermark strength of DCT coefficients. The experimental results showed that the method could increase the watermarking strength and the robustness of the watermark was enhanced. Furthermore, Jin and Wang (2007), indicated that using ANN in different textural features of each DCT block, the luminance of an image can be implored to decide adaptively the watermarking-embedding strength. Similarly, Ming et al, (2003) defined a Radial Basis Function neural networks-based algorithm that controlled and created a maximum image-adaptive strength watermark.

Artificial Neural Network in Error Rate Prediction

Some authors have used neural network to predict the error rate of watermarked images. The watermark-detection error rate helps embed more watermark

messages while keeping an acceptable detection error rate. It is useful in the design of general watermarking and detection algorithms. According to the experimental results obtained by Zhang et al. (2007), 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 the 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 et al. (2008), proposed a frequency-based transform watermarking using NN on the YCbCr domain to detect a hidden code from the content. A conditioned neural network was used as a classifier to recognize a hidden bit pattern from the content which the embedder associated to the target content. They reported that the method did not damage the target content. However, the extraction keys had to be shared among the embedder and the extractor in order to extract the hidden bits completely from the target content.

Artificial Neural Network in Tamper Detection

Another area of watermarking that researchers have looked into is in tamper detection phase. Detection is to find if the watermark is present or not. During the watermark recovery, just before the extraction, a detection threshold may be set to establish whether a watermarked media contains a mark or not Only two values are set true or false/ 1 or 0. Usually the correlation test method is used in watermark detection. There are two types of error detection (Cox et al., 1999); the false negative, when the watermark detector fails to detect a watermark that is present, and False positive, when the watermark detector indicates the presence of a watermark in an unwatermarked media.

Artificial Neural Network in Location of Safe Region

Recently, a new watermarking component was found; the Secure Region (SR) within the capacity estimated. According to Olanrewaju et al., 2009, Secure Region (SR) is an identified region in the host media in which when the watermark is hidden therein, it will not be destroyed nor degraded. ANN is used to locate such a region (Olanrewaju et al., 2010).

So far, to the best of our knowledge, no work has been done on watermarking using Complex- Valued Neural Network (CVNN). All of the above algorithms used Real-Valued Neural Network (RVNN), that is, the input to the network is real values and the output of the network is real values as well. The CVNN method enables the network learn complex numbered patterns naturally. The

input, weights and output of the network are complex valued. Traditionally, watermarks are embedded to the phase or magnitude (real values) of the host image in the DFT domain. However, embedding in phase only (Cheng & Cheng, 2005) or magnitude only (Hernandez et al, 2005) may lead to loss of information. With CVNN, both components can be used for embedding simultaneously without the loss of information. Therefore, neural network dealing with input and output vectors expressed in complex numbers are strongly desired, such as in optoelectronics, imaging, remote sensing, etc., (Liu et al 2009).

In this paper, a new scheme for embedding watermark into digital images using CVNN is presented. The proposed algorithm is operated in the Fast Fourier Transform (FFT) domain.

COMPLEX-VALUED NUERAL NETWORK SYSTEM

Complex numbers are used to express real-world phenomena such as in a image that comprises of phase and magnitude. However, to process such signals by neural network, a Complex-Valued Neural Network (CVNN) is needed (Amin & Murase, 2009). When using the conventional Real-Valued Neural Networks (RVNN), one must apply the method individually to the real and the imaginary part of a signal. On the other hand, CVNN allows direct processing of both the real and the imaginary component of a signal simultaneously. Moreover, CVNN enables us to capture the good rotational behavour of a complex number (Nitta, 2003). CVNN as shown in Figure 3 consists of the Complex-Valued Feed Forward (CVFF) and the Complex Back-Propagation (CBP) algorithms. CVNN has been studied and developed by Leung & Haykin (1991), Georgiou. & Koutsougeras (1992), Kim & Adali (2001), Hanna & Mandic (2002), and Kim and Adali (2002). The Complex-Valued Neural Network (CVNN) consists of an interconnection of the Complex-Valued (CV) neurons and Complex-Valued synaptic weights as depicted in Figure 4. It processes information using a connectionist approach to computation in the complex domain. CVNN starts by transmitting the complex input signals or data through the connection; each connection has an associated weight that improves the transmitted signal; each neuron transforms the received signals (sums the input multiplied by the connection weight as in equation (2) through an activation function which in turns determines the output signal.

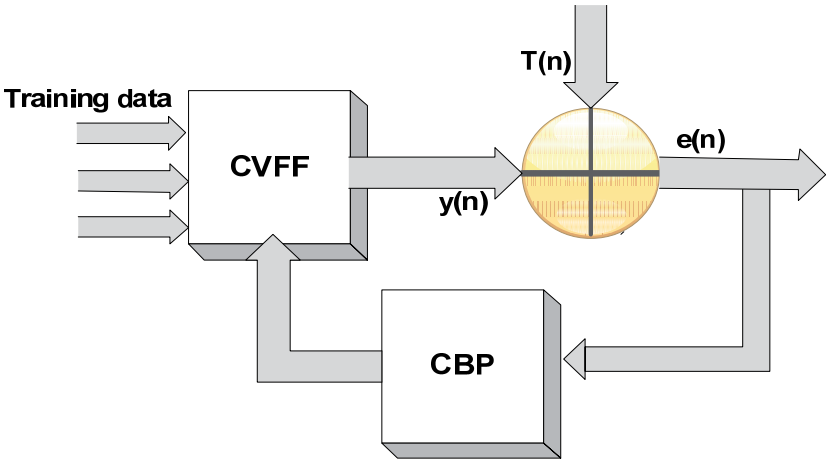

Figure 3. CVNN; Consisting of the CVFF and the Complex Back Propagation Algorithm

2n this Fifure $T(n)$ is the desired output, $y(n)$ is the actual output of CVNN, $e(n)$ is the error to be propagated backward.

Complex Neuron

A model of the complex neuron used in this study as shown in Figure 4, is divided into two parts: the summation, and the activation function parts. It begins by summing up the weighted complex-valued inputs in order to obtain the threshold value which will be used to represent the internal state of a given input pattern. The net output y_n of a complex-valued neuron n is shown by:

$$y_n = \sum_{m=0} X_m W_{nm} \qquad (2)$$

where W_{nm} is a complex synaptic weight connecting neuron n and m, X_m is the complex input signal from neuron m. However y_n is a function of z; $f(z)$ isthe activation function. Details of the activation function are discussed in section 3.2.

All the complex inputs are computed based on the complex algebra which results into a complex output through complex weights. The resultant sum is fed into the activation function which maps the weighted sum to the real value output. The complex-value neurons are used in both the hidden and the output layers.

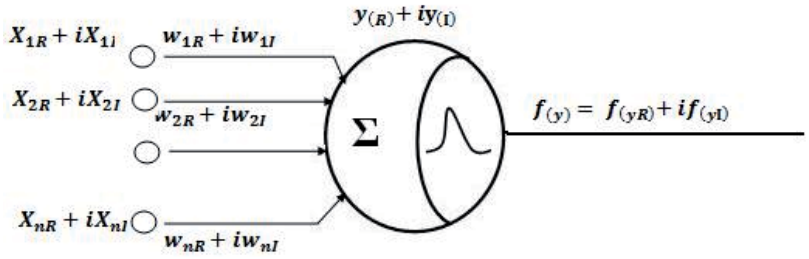

Figure 4. Complex Neuron Model

Activation Function

The neuron uses the sigmoid activation function which is applied to both the real and the imaginary parts separately. As soon as the result from the summation part is fed into the activation function part, the activation function is triggered which causes the output to be bounded. For this study, the sigmoid activation function is used. It is defined as:

$$y_{n(R)} = \frac{1}{1 + exp^{-\tau * real\,(x)}} \tag{3}$$

$$y_{n(I)} = \frac{1}{1 + exp^{-\tau * imag\,(x)}} \tag{4}$$

$$z = y_{n(R)} + iy_{n(I)} \tag{5}$$

The error is calculated to be

$$e_{(n)} = \left[T_{n(R)} + iT_{n(I)}\right] - y_{n(R)} + iy_{n(I)} \tag{6}$$

Where $e_{(n)} = T_{n(R)} + iT_{n(I)}$ is the target complex valued data and is the output of CVNN. The objective of CVNN training using the complex backpropagation technique is to find a set of parameters that minimize the sum of the squared of the error function, that is

$$E_{(n)} = \frac{1}{2}\sum^{l} e_{(n)} * e_{(n)} \tag{7}$$

Where $* e_{(n)} = * e_{n(R)}$ - is the complex conjugate of the error function, l is the total number of neurons in the output layer. For the CVNN, $l = 1$. During the training, the complex network weight update is given by:

$$w(n+1) = w(n) + \Delta w(n+1) \tag{8}$$

But

$$\Delta w = -\mu \nabla_w E_n \tag{9}$$

where μ is the learning rate, E_n is the gradient of the cost function and w is a complex weight function with Real $_{(R)}$ and Imaginary $_{(I)}$ parts. Therefore, (9) can be re-written as a complex variable with partial derivatives as:

$$\mu \nabla_w E_n = \frac{\partial E_n}{\partial w_{njR}} + i \frac{\partial E_n}{\partial w_{njI}} \tag{10}$$

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 needs to be taken into consideration. The partial derivatives can be written as:

$$\frac{\partial E_n}{\partial w_{njR}} = \frac{\partial E_n}{\partial y_R}\left(\frac{\partial y_R}{\partial u_R}.\frac{\partial u_R}{\partial w_R} + \frac{\partial y_R}{\partial u_I}\frac{\partial u_I}{\partial w_{njR}}\right) + \frac{\partial E_n}{\partial y_I}\left(\frac{\partial y_I}{\partial u_R}.\frac{\partial u_R}{\partial w_R} + \frac{\partial y_I}{\partial u_I}\frac{\partial u_I}{\partial w_{njR}}\right) \tag{11}$$

$$\frac{\partial E_n}{\partial w_{njI}} = \frac{\partial E_n}{\partial y_R}\left(\frac{\partial y_R}{\partial u_R}.\frac{\partial u_R}{\partial w_I} + \frac{\partial y_R}{\partial u_I}\frac{\partial u_I}{\partial w_{njI}}\right) + \frac{\partial E_n}{\partial y_I}\left(\frac{\partial y_I}{\partial u_R}.\frac{\partial u_R}{\partial w_I} + \frac{\partial y_I}{\partial u_I}\frac{\partial u_I}{\partial w_{njR}}\right) \tag{12}$$

Finding the partial derivatives associated with (11) - (12) and applying the Cauchy- Reima condition leads to the CVNN weight update given as

$$\begin{aligned} \nabla_w E_n &= -w(n) + \beta \bar{X}_{jn} \bar{F}'\big(u(n)\big)\partial(n) \\ &= -\bar{X}_{jn} \bar{F}'\big(u(n)\big)\partial_R(n) + i\partial_{uI} \\ &= -\bar{X}_{jn} \bar{F}'\big(u(n)\big)\partial(n) \end{aligned} \tag{13}$$

The complex network weight update for a fully complex CVNN is thus given as:

$$\begin{aligned} w(n+1) &= w(n) + \beta \bar{X}_{jn} \bar{F}'\big(u(n)\big)\partial(n) \\ &= w(n) + \mu \bar{X}_{jn}\, \partial_R(n) F'(u_R) + i\bar{F}'\, i\partial_I(n) F \end{aligned} \tag{14}$$

METHODOLOGY

The block diagram of the proposed Damageless watermarking system is shown in Fig. 6. It consists of a three-stage cascade system, namely Fast Fourier Transform (FFT), CVNN and the Embedding section. Details are discussed in subsection 4.1 to 4.3

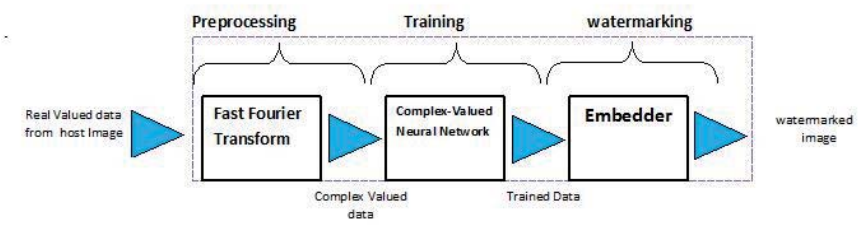

Figure 5. Block Diagram of Damageless Watermarking System

Discrete Fourier Transform; Some Complex Values

Block decomposition and transformation of the host image from the spatial to the transform domain using the fast version of Discrete Fourier Transform (DFT) is the first step in the FFT process. Given an image *I(x, y)* of size *MXN*, for $x = 0, 1, \ldots M-1$ and $y = 0, 1 \ldots N-1$, the 2-D DFT of *I(x, y)* is represented by *F*(*u, v*):

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} I(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad (15)$$

Thus, given *F*(*u, v*), we can obtain *I*(*x, y*) back by means of the Inverse 2-Dimensional DFT (2D IDFT).

$$I(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})} \quad (16)$$

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

Figure 6 shows how FFT is used to transform image *I(x,y)* from the spatial domain to the transform domain in order to obtain complex values.

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(x,y)* with size *MxN*, there should be N^2 FFT points such as 16, 64, 128, 256, etc. This is because the FFT points must not be smaller than the data length. For this study, a FFT point of 16 is used. The FFT shift of each block was then taken to determine the DC component. Once the data are sorted, the midpoint is easily located. This procedure helps in partitioning the data into the left-hand side (LHS) and the 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 data on the LHS is negative and its target is 1. Further categorization is done on both RHS and LHS based on the frequency component to low, mid and high frequencies. These serve as the final complex input data to be fed into the CVNN.

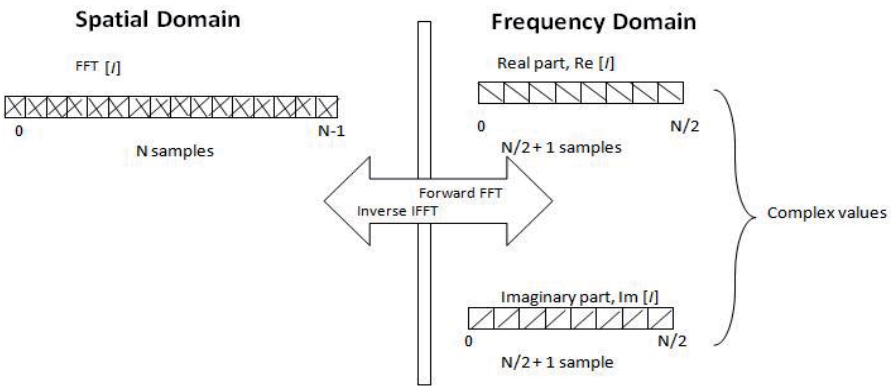

Figure 6. Conversion of image from spatial domain to transform domain in order to obtain complex value using FFT (adapted from Smith, 1999).

CVNN Training

After the FFT values of the host image were calculated which contain the real and the imaginary components as explained in section 4.1, these data were used as the input data for the CVNN. Figure 7 show the real and the imaginary parts (complex values) of the host image. At the beginning of the training, weights and bias were initialized with small random complex values. After several experimentations, the optimum architecture (shown in Figure 8) for the CVNN is found to be 11:6:1: 11 nodes at the input layer, 6 nodes at the hidden layer and 1 node at the output layer with a 0.69 learning rate.

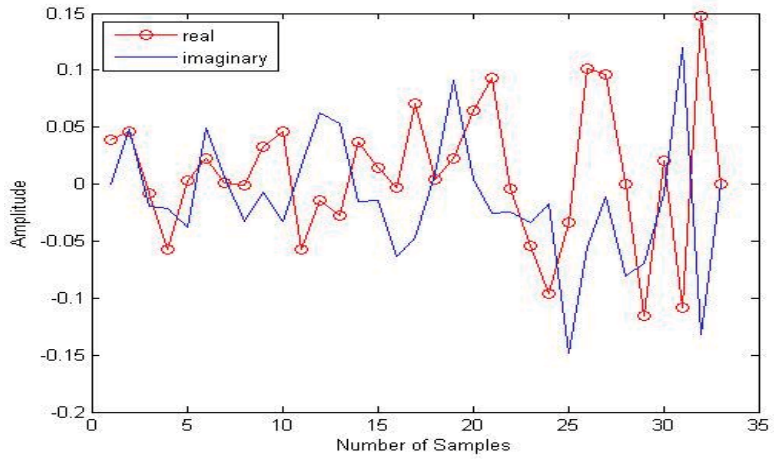

Figure 7. Real and Imaginary Components of FFT Result for Sub-block of Host Image(Block9)

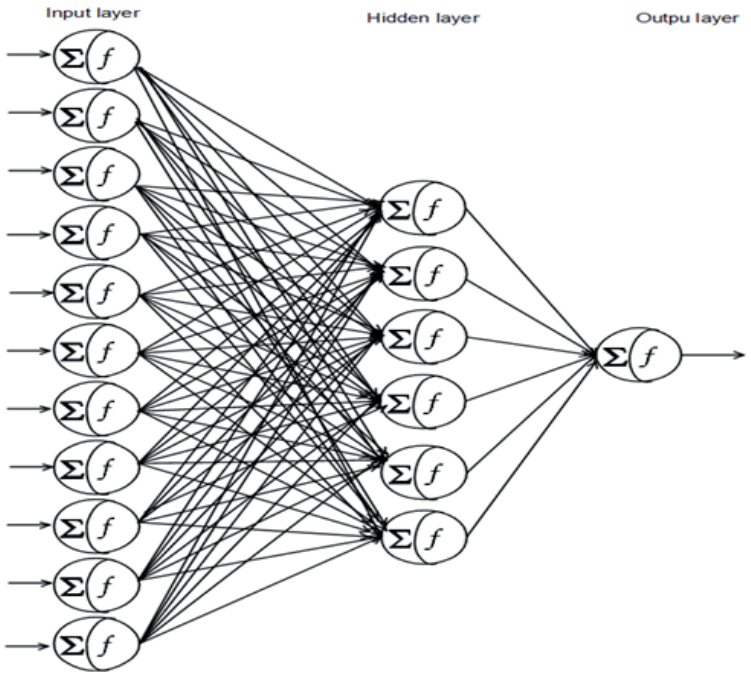

Figure 8. Damageless Watermarking CVNN Network Topology

Learning Convergence

Generally when any neural network is trained with inputs, the error on training dataset decreases gradually with the epochs or the goal set. The CVNN stops training when the goal is achieved or reached or the maximum number of epochs are specified; which-ever condition is met first. Equation (17) was used as the criteria to stop the training when the maximum epoch set was achieved (1500epochs). Once the condition in Equation (18) was satisfied, the algorithm stopped training, weights and bias were blocked. Figure 9 shows how the mean square error changed over the epochs.

$$E_{(n)} = \frac{1}{2J}\sum_{j=1}^{k}\sum_{n=1}^{l}(T_{jn} - y_{jn})^2 \tag{17}$$

$$E_{(n)} = \frac{1}{2J}\sum_{j=1}^{k}\sum_{n=1}^{l}(T_{jn} - y_{jn})^2 = 1500 \tag{18}$$

where T and y are complex numbers representing the target and the output, J is the total number of training patterns, k is the pattern number while l is the number of output neurons.

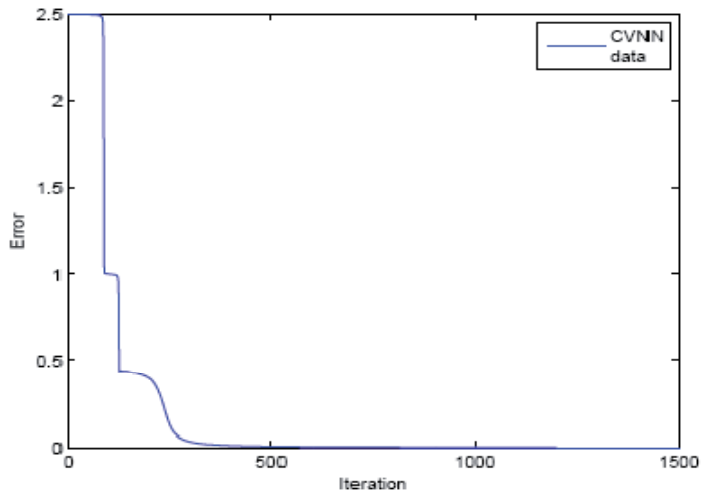

Figure 9. A Typical Mean Square Error, MSE Plot for CBP Algorithm

Watermark Vectorising and Binarization

The watermark is a 19x17 image of the IIUM logo as shown in Figure 10. It was vectorized and the gray level was calculated for binarization. The resultant vector serves as the target or classification signal. The size of the watermark determines the number of blocks to be selected for CVNN training data. The watermark also determines the class type each watermark bit will be mapped to, either positive or negative. 0 bit is embedded in the positive while 1 bit is embedded/mapped to the negative part. For a binary sequence $b_{(m,n)}$ of watermark $w_{(m,n)}$ is mapped to the RHS negative, or the LHS positive of the frequency spectrum by applying the following constraint:

$$b_{(m,n)}\left\{\begin{matrix}1\\0\end{matrix}\middle| f_{(m,n)=\begin{matrix}-ve\\+ve\end{matrix}}\right\} \tag{19}$$

Where $f_{(m,n)}$ is the frequency spectrum

$$w = \{w_{(m,n)} | w_{(m,n)} \epsilon \{0,1\}\} \tag{20}$$

Embedding Method

The main step in embedding is sufficient generation of weights (hidden and output) by the CVNN and carefully mapping of the target content (watermark) to the input data (host image). Our method uses the mapping strategy to embed the watermark instead of the traditional adding (+) of the watermark to the host image. It is damageless because there is no physical addition (+) of the watermark to the host image during embedding. Each block is mapped with a watermark bit as illustrated in Figure 10, the embedding/mapping procedure.

1. Block selection: Consider a grayscale host image *I(x, y)* 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(x, y)* is x 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 by FFT to obtain the complex values for the CVNN is explained in section 4.1.

3. Establishing a CVNN relationship between the input data and the target content, which is, carefully matching the target content with the input data. A three bits binary target content *T* is mapped to the input data. The mapping rule is determined by:

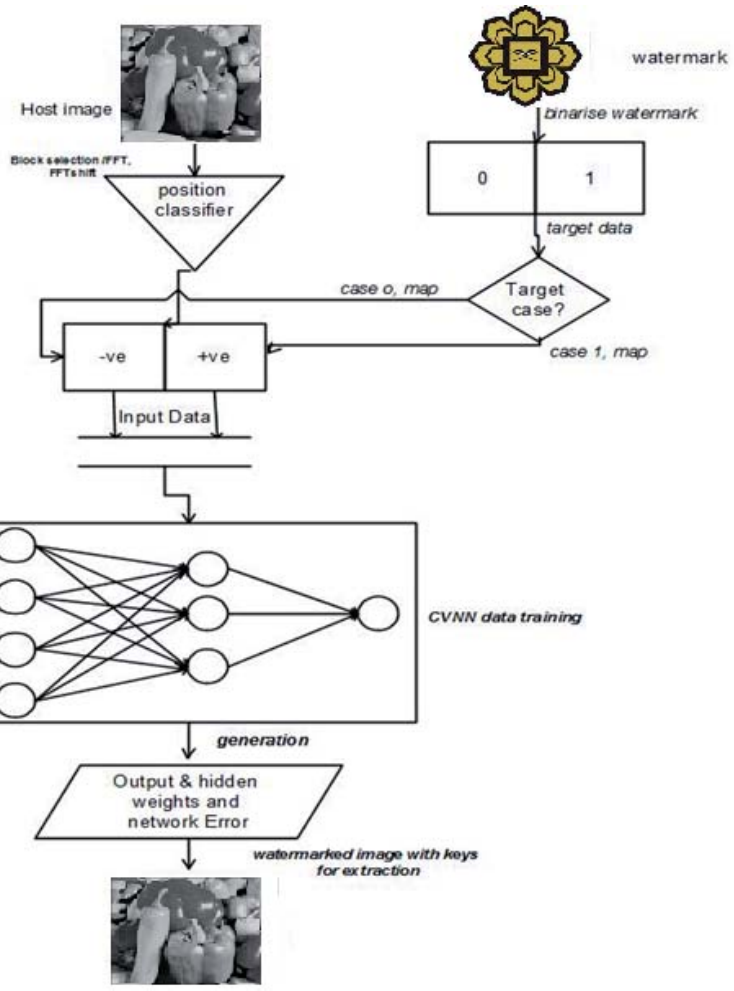

Figure 10. The Watermarking Embedding Procedure

$$T = \{t_{(m,n)} | t_{(m,n)} \epsilon \{0,1\}\} \tag{21}$$

Where $t\,(m,n)$ is defined as follows:

$$t_{(m,n)} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{matrix} | \\ | \end{matrix} \begin{matrix} mid \\ otherwise \end{matrix} \tag{22}$$

Where T is the target content and is the position of the low, mid and high frequencies.

4. Training is repeated and the relationship is adjusted between the target content and the corresponding output of the CVNN model until the network learning threshold is satisfied and the convergent of network weights are achieved.

5. Weights are saved for future extraction.

Figure 11. IIUM Logo as the Watermark

Figure 12. Pepper, the Host Image

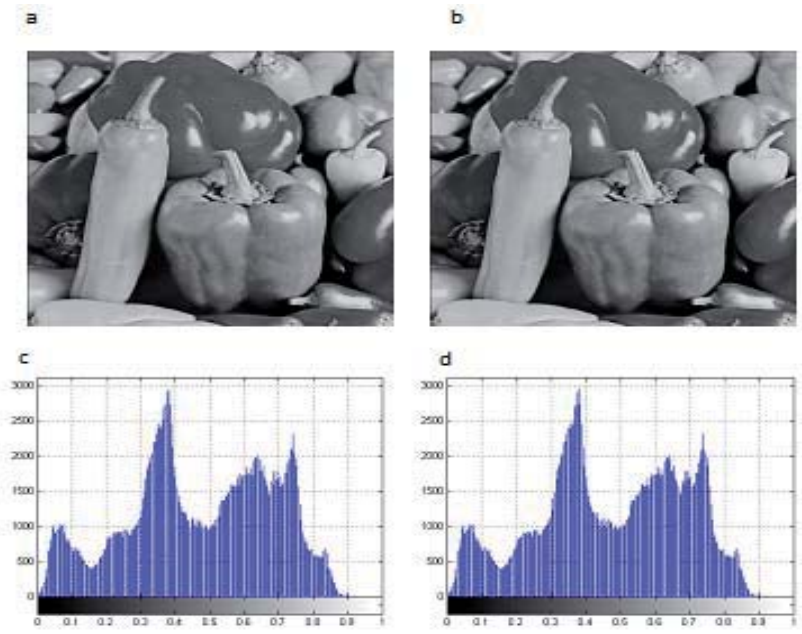

Figure 13a. Host image, Pepper, b. The Watermarked Pepper, c. Histogram of Host Image d. Histogram of Watermarked Imaged.

RESULT AND DISCUSSION

Figure 13b shows the result of the watermarked image and its corresponding histogram is shown in 13d. Since we do not embed any data into the cover image, there will be no visual quality degradation to the watermarked image, which fulfills the imperceptibility requirement of watermarking algorithm. As can be seen from Figures 13c and 13d, the histograms of the host image and the watermarked image show that both images are identical. This imperceptible result is achieved due to the mapping strategy. Here, we would like to stress more on the robustness of the proposed algorithm especially the extraction of the watermark bits.

Table 1 shows the result of a single run for training the embedder using a three bit watermark on an average of 40 sub blocks. Results of the preliminary experiment showed that the algorithm was able to map 3 bits of watermark data per one sub-block. The watermark pattern for all the sub-block frequencies (low, mid and high) are three bits, 0 1 0. 40% of the original data was used for training the network. Once the network was correctly trained, another separate

40%, different from the training data, was used as test data and the remaining 20% for validation. These data were taken from independent blocks. For the test result for low frequency neuron output, using the saved weights in Table 2, the watermark were correctly mapped and recovered, which showed that the weights used for extraction were sufficiently enough to recover the watermark bit accurately. For all the 0 bits mapped, the test result was between 0.00 and 0.04, which is considered a very good extraction. This is because after taking the threshold, the values fall back to 0. The results buttress the point that watermark signals were recovered only when the correct input data was used. For the mid frequency neuron output in column 3, the mapped bit is 1. The result after extraction gives between 0.99 and 0.95. This is also an indicator that the watermark is correctly recovered; 0.99 is considered 1 after the threshold. The high frequency neuron in column 4 output is mapped to 0. The result after extraction shows retrieved values between 0.00 and 0.04. This is also a good output because it recovered the exact bit after the tresholding in column 5. The result also shows that there were no wrongly mapped bit. This indicates that the algorithm is very accurate. Objective test for accuracy is future work.

Regarding learning convergence, the algorithm converges before the training cycles(epoch) are completed, which is 500 epochs instead of the 1500 set epoch. This is evidence that it takes less time to reach the minimum validation error. Figure 9 show a typical training cycle.

EVALUATION CRITERIA

The performance of the algorithms were appraised by some objective performance measures such as: Mean Square Error (MSE) and Average Difference Indicator (ADI). MSE and MDI were used as accuracy indicators for the retrieved watermark. Technically, MSE and ADI are similari measurements between two different signals. The value ranges between 0 and 1. When the result of two signals/images is 0 it means they are similar while 1 means dissimilar; that is, a lower value signifies closeness.

MSE and ADI are defined as:

$$\text{MSE}= \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}|U(m,n)-V(m,n)|^2 \qquad (23)$$

$$\text{ADI}= \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(U(m,n)-V(m,n)) \qquad (24)$$

Table 1

Watermarked Neuron Output and Performance Evaluation.

	Frequency component of recovered watermark.				Measure of quality	
Embedded watermark	**Low**	**Mid**	**High**	**Threshold (x>0.5) of Recovered watermark**	**MSE**	**ADI**
010	0.0355	0.9638	0.0021	**010**	0.01	0.00
010	0.0226	0.9768	0.0084	**010**	0.01	0.01
010	0.028	0.9686	0.0169	**010**	0.01	0.02
010	0.0355	0.9658	0.0002	**010**	0.01	0.00
010	0.032	0.9707	0.0012	**010**	0.01	0.00
010	0.0267	0.9786	0.0022	**010**	0.00	0.00
010	0.0223	0.9759	0.0125	**010**	0.01	0.01
010	0.0283	0.9707	0.0029	**010**	0.01	0.00
010	0.0307	0.9683	0.0132	**010**	0.01	0.01
010	0.0268	0.9695	0.016	**010**	0.01	0.02
010	0.0397	0.9574	0.0068	**010**	0.02	0.01
010	0.0329	0.9633	0.0139	**010**	0.01	0.01
010	0.0334	0.9658	0.002	**010**	0.01	0.00
010	0.0142	0.9879	0.0025	**010**	0.00	0.00
010	0.0267	0.9641	0.0194	**010**	0.01	0.02
010	0.0257	0.9556	0.0252	**010**	0.01	0.02
010	0.0313	0.9641	0.0165	**010**	0.02	0.01
010	0.0373	0.9603	0.013	**010**	0.01	0.01
010	0.0205	0.9739	0.0129	**010**	0.00	0.00
010	0.0139	0.9858	0.0001	**010**	0.01	0.03
010	0	0.9636	0.0319	**010**	0.03	0.05
010	0.0024	0.9472	0.0451	**010**	0.01	0.03
010	0.0018	0.9674	0.0322	**010**	0.00	0.01
010	0.0163	0.9783	0.0132	**010**	0.01	0.03
010	0	0.9668	0.0279	**010**	0.01	0.03
010	0.0007	0.9745	0.0261	**010**	0.01	0.03
010	0.001	0.9733	0.029	**010**	0.01	0.03
010	0.0036	0.9679	0.029	**010**	0.01	0.03
010	0	0.9721	0.0257	**010**	0.03	0.05
010	0.0003	0.9473	0.047	**010**	0.01	0.03
010	0.0017	0.9681	0.0258	**010**	0.01	0.03
010	0.001	0.9668	0.0326	**010**	0.01	0.03
010	0.0016	0.9726	0.0309	**010**	0.01	0.03
010	0.0028	0.9673	0.0343	**010**	0.01	0.04
010	0	0.9628	0.0356	**100**	0.02	0.04
010	0.0011	0.9523	0.0446	**010**	0.02	0.04
010	0.0015	0.9544	0.0438	**010**	0.01	0.03
010	0	0.9726	0.0308	**010**	0.01	0.04
010	0.0041	0.9667	0.0353	**010**	0.01	0.03

From the result obtained in Table 1 column 6, the mapped bit/target bit and the recovered bit for 40 independent blocks show that MSE is between 0.00 - 0.03, which is considered to be highly similar by its definition, that is, 0.03 is closer to 0 than to 1. For ADI, the obtained evaluation result for the mapped bit and the recovered bit is between 0.00 - 0.05. Apparently this shows that the mapped bits are very similar to the recovered watermark. Figure 12 shows the variation of the mid frequency extracted watermarks in relation to the embedded watermarks.

Table 2

Learned Hidden and Output Weights of CVNN for 11 Inputs

Complex Hidden Weights (real and imaginary components)											
0.7098 + 0.0231i	0.7608 - 0.1085i	1.2545 + 0.1907i	1.2837 + 0.2815i	0.7733 + 0.1017i	0.3700 - 0.1149i
0.6379 - 0.0370i	0.9017 + 0.1094i	0.1082 - 0.1392i	0.8171 - 0.2010i	0.0576 - 0.0604i	0.7149 + 0.1014i
-0.7209 -0.0538i	0.0556 + 0.1789i	0.5707 - 0.2451i	0.8940 - 0.3662i	0.7862 - 0.0974i	0.8296 + 0.1666i
0.7155 + 0.0398i	1.0945 - 0.1142i	0.1304 + 0.1577i	-0.0812 - 0.2380i	0.3442 + 0.0657i	0.7259 - 0.1066i
0.9307 + 0.0834i	-0.2489 - 0.2027i	0.6664 + 0.2891i	1.4619 + 0.4331i	0.2484 + 0.1222i	0.3208 - 0.1961i
0.5096 - 0.0505i	0.4679 + 0.1465i	0.1490 - 0.1945i	0.5415 - 0.2865i	0.1953 - 0.0827i	0.9433 + 0.1367i
0.1894 - 0.0103i	0.2749 + 0.0849i	0.1703 - 0.0963i	1.0867 - 0.1310i	0.3160 -0.0362i	0.4552 + 0.0741i
0.3781 + 0.0643i	0.3833 - 0.1889i	0.4506 + 0.2489i	0.4192 + 0.3663i	0.2306 + 0.1061i	0.3264 - 0.1754i
0.8155 + 0.0215i	0.1097 - 0.0641i	0.2358 + 0.0787i	0.5903 + 0.1137i	0.7004 + 0.0355i	0.9581 - 0.0582i
0.2581 - 0.0388i	0.1846 + 0.2200i	1.1643 - 0.2369i	0.4950 - 0.3185i	0.8899 - 0.1002i	0.2201 + 0.1898i
0.3190 - 0.0041i	0.6477 + 0.0056i	0.4860 - 0.0116i	0.8630 - 0.0190i	0.5285 - 0.0046i	0.6326 + 0.0065i
Complex Output Weights (real and imaginary components)					
-0.4441 – 1.0333i	1.3803 - 1.9589i	-1.3612 + 0.8452i	-2.0742 + 1.3051i	-0.3152 + 0.5714i	1.1096 - 0.5664i

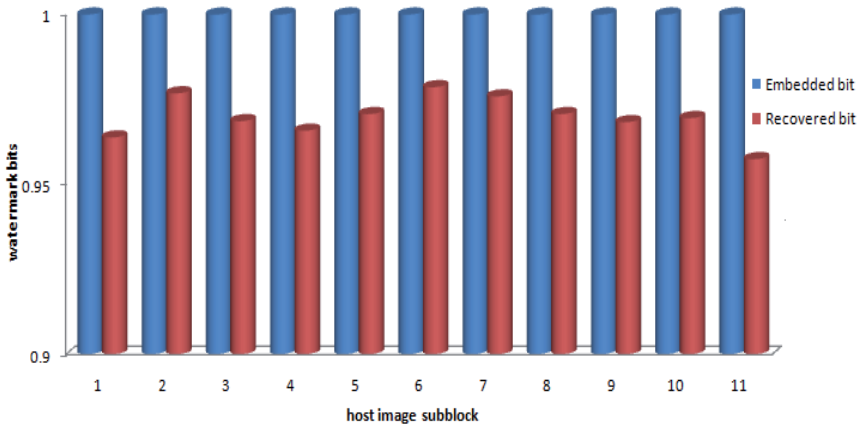

Figure 14. The Midfrequency Embedded Watermark Bits and the Respective Extracted Bits

CONCLUSION

In this paper, a new method of embedding a Damageless watermark based on CVNN has been presented. The method is damageless in the sense that there is no physical "touch" of the host image. The embedding strategy is mapping-based which does not require the watermark to be traditionally added (+) to the host image. The method converts a real domain data to a complex domain via Fourier transform techniques to obtain complex values. The transformed data was then used to train a CVNN system. Once the network was properly trained it was tested on some other cases that were unknown to the system. The effectiveness of the algorithm was evaluated using the objective performance measure and it showed that the algorithm was able to recover all the embedded bits acurately. The first major contribution of this work was is the innovative application of CVNN in embedding. In the field of signal and image processing where inputs, outputs and transfer function are modelled in the complex domain, this proposed CVNN-based embedder will be a useful tool for watermarking. The second contribution was the level of accuracy obtained. Increasing the watermark bits and attacking the watermarked image is enumerated as a future work problem.

REFERENCES

Abdallah, E. E., Hamza, A. B., & Bhattacharya, P. (2006). A robust block-based image watermarking scheme using fast hadamard transform and singular value decomposition. *IEEE International Conference on Pattern Recognition*, *3*, 673–676.

Aibinu, A. M., Salami, M. J. E., Shafie, A. A., & Najeeb, A. R. (2008). Increasing the accuracy and speed of convergence of an ARMA coefficient determination using neural network technique. *World Assembly of Engineers, Scientists and Technologists International Conference on Computer System Engineering* (ICCSE), *32*, 196–202.

Amin, M. F., & Murase K. (2009). Single-layered complex-valued neural network for real-valued classification problems. *Neurocomputing* , 72, 945–955.

Bansal A. E., & Bhadauria, S. S. (2005). Watermarking using neural network and hiding the trained network within the cover image. *Journal of Theoretical and Applied Information Technology,* 663–670.

Bender, W., Gruhl, D., Morimoto, N., &. Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, *35*, (3/4), 313–336,

Celik, M. U., Lemma, A. N., Katzenbeisser, S., & Veen, M. V. (2008). Lookup-table-based secure client-side embedding for spread-spectrum watermarks. *IEEE Transactions on Information Forensics and Security, 3*, (3), 475–487.

Chang, C. Y. (2005). The application of a full counterpropagation neural network to image watermarking. *Networking, Sensing and Control*, 993–998.

Chen, W. -Y., & Chen, C. -H. (2005). A robust watermarking scheme using phase shift keying with combination of amplitude boost and low amplitude block selection. *Pattern Recognition*, *38*, (4), 587–598.

Choi, H. J., Seo, Y. H., Yoo, J. S., & Kim, D. W. (2008). Digital watermarking technique for holography interference patterns in a transform domain. *Optical and Lasers in Engineering, 46*, (4), 343–348.

Cox, I. J., Miller, M. L., & Bloom, J. A. (1999). *Digital watermarking.* Morgan Kaufmann.

Gao, T., & Gu, Q. (2007). Reversible image authentication based on combination of reversible and LSB algorithm. *Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, 4425576, 636–639.

Ge, S., Gao, Y., & Wang, R. (2007). Least significant bit steganography detection with machine learning techniques. *Conference on Knowledge Discovery in Data, Proceedings International Workshop on Domain Driven* **,** 24–32.

Georgiou, G., & Koutsougeras, C. (1992). Complex domain backpropagation. *IEEE Trans. On Circuits and Systems II*, *39*, (5), 330–334.

Ham, F. M., & Kostanic, I. (2002). *Principles of neurocomputing for science & engineering*. Singapore: Mc.GrawHill, 136–140.

Hanna, A. I., & Mandic, D. P. (2002). A normalised complex backpropagation algorithm. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing,* 1, 977–980.

Haykin, S. (2008). *Neural networks and learning machines (*3rd ed.*)*. Prentice Hall.

Jin, C., & Wang, S. (2007). Applications of a neural network to estimate watermark embedding strength. *IEEE 8th International Workshop on Image Analysis for Multimedia Interactive Services*, 68–68.

Hernandez, M. C., Miyatake M. N., & Meana, H.M.P. (2005). Analysis of a DFT-based watermarking algorithm. *2nd International Conference on Electrical and Electronics Engineering,* 44–47.

Khan, A., Tahir, S. F., Majid, A., & Choi, T. S. (2008). Machine learning based adaptive watermark decoding in view of anticipated attack. *Pattern Recognition, 41*, 2594–2610.

Kim, T., &. Adali, T. (2002). Complex backpropagation neural network using elementary tran-scendental activation functions. *Proceedings of IEEE ICASSP*, 2.

Leung, H., & Haykin, S. (1991). The complex backpropagation algorithm. *IEEE Trans. On Signal Proceedings 3*, (9), 2101–2104.

Li, X., & Wang, R. (2007). A video watermarking scheme based on 3D-DWT and neural network. *Ninth IEEE International Symposium on Multimedia*, 110–115.

Liu, X., Fang, K., and Liu, B. (2009). A synthesis method based on stability analysis for complex-valued Hopfield neural network. *Proceedings of the 7th Asian Control Conference*. Hong Kong, China, 1245–1250.

Lu, W., Sun, W., & Lu, H. (2009). Robust watermarking based on DWT and nonnegative matrix factorization, *Computers & Electrical Engineering*, 35, 183–188.

Luo, X., Liu, B., & Liu, F. (2005). Improved RS method for detection of LSB steganography. *Lecture Notes in Computer Science, Computational Science and its Applications.*

Majhi, B., & Shalabi, H. (2005). An improved scheme for digital watermarking using functional link artificial neural network. *Journal of Computer Science*, *1*, (2), 169–174.

Me , S. -C., Li, R. -H., Dang, H. –M., & Wang, Y. -K. (2002). Decision of image watermarking strength based on artificial neural-networks. *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'O2), 5,* 2430–2434.

Ming, Z. Z., Rong-Yon, L., & Le, W. (2003). Adaptive watermark scheme with RBF neural networks. *IEEE International Conference Neural Networks & Signal Processing*, *2*, 1517–1520.

Naoe, K., & Takefuji, Y. (2008). Damageless information hiding using neural network on YCbCr domain. *International Journal of Computer Science and Network Security*, *8* (9).

Nita, T. (2003). Orthogonal decision boundaries and generalization of complex-valued neural networks. I Complex-Valued Neural Networks: Theories and applications, (Ed). Akira Hirose, World Scientific., 7–28.

Olanrewaju, R. F., Khalifa, O. O., Abdalla, A., Aburas, A. A., Zeki, A. M. (2010). Watermarking in safe region of frequency domain using complex-valued neural network. *Proceedings of International Conference on Computer and Communication Engineering*.

Olanrewaju, R. F., Aburas, A. A, Khalifa, O. O., & Abdalla, A. (2009) State-of-the-art application of artificial neural network in digital watermarking and the way forward. *International Conference on Computing and Informatics,* 233–237.

Palit, A. K., & Popovic, D. (2005). *Computational intelligence in time series forecasting*. Springer.

Parthasarath, A. K., & Kak, S. (2007). An improved method of content based image watermarking. *IEEE Transactions on Broadcasting, 53*, (2).

Sang, J., & Alam, M. S. (2008). Fragility and robustness of binary-phase-only-filter-based fragile/semi fragile digital image watermarking. *IEEE Transactions on Instrumentation and Measurement*, *57*, 3, 595–606.

Schmidt, T., Rahnama, H., & Sadeghian, A. (2008). A review of applications of artificial neural networks in cryptosystems, *World Automation Congress,* 1–6.

Senthil, V., Bhaskaran, R. (2007). Wavelet based digital image watermarking with robustness against geometric attacks. *International Conference on Computational Intelligence and Multimedia Applications, 4*, 89–**93.**

Servetto, S. D., Podichuk, C. I., & Tamachandran, K. (1998). Capacity issues in digital image watermarking. *International Conference on Image Processing, 1*, 445–449.

Shi-Chun, M., Ren-Hou, L., Hong-Mei, D. & Yun-Kuan, W. (2002). Decision of image watermarking strength based on artificial neural-networks. *IEEE Proceedings of the 9th International Conference on Neural Information Processing*, 5, 2430–2434.

Shih Y. F., & Wu, Y. T. (2005). Robust watermarking and compression for medical images based on genetic algorithms. *International Journal of Information Sciences*, 175, 200–216.

Smith, S. W. (1999). The scientist and engineer's guide to digital signal processing. Retrieve from http://www.dspguide.com/ch8/6.htm

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.

Tsai, H. H. (2007). Decision-based hybrid image watermarking in wavelet domain using HVS and neural networks. In D. Liu et al. (Eds.): *ISNN*, Part III, 4493, 904–913.

Van., R. G., Tirkel, S. A. Z., & Osborene, C. F. (1994). A digital watermark. *In Proceedings IEEE International Conference Image Processing*, 2, 86–92.

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.

Wong, H. W. P., & Au, O. C. (2003). A capacity estimation technique for JPEG to JPEG image watermarking. *IEEE Transactions on Circuits and Systems for Video Technology, 13*,8 746–752 .

Wu, M., & Liu B. (2003). Data hiding in image and video: Part I—fundamental issues and solutions. *IEEE, Transactions on Image Processing, 12*, 16. 685–695.

Zain, J. M., & Fauzi Abdul, R.M. (2007). Evaluation of medical image watermarking with tamper detection and recovery (AW-TDR): *Proceedings of the 29th Annual International Conference of the IEEE EMBS,* 5661–5664, Lyon France.

Zhang, F., & Zhang, H. (2005). Applications of a neural network to watermarking capacity of digital image. *Neurocomputing*, *67,* 345–349.

Zhang, F., Zhang, X., & Zhang, H. (2007). Digital image watermarking capacity and detection error rate. *Pattern Recognition Letters*, *28*, 1–10.

Zhang, J., Wang, N., & Xiong, F. (2002). Hiding a logo watermark into the multiwavelet domain using neural networks. *Proceedings of 14th IEEE International Conference on Tools with Artificial Intelligence, (ICTAI 02)*, 477–482.

Zhang, X. H., & Zhang, F. (2005). A blind watermarking algorithm based on neural network, *International Conference on Neural Networks and Brain*, *2*, 1073–1076.

Kuttera, M., and Petitcolas, F. A. P. (2000). Fair evaluation methods for image watermarking systems. *Journal Electron Imaging*, *9*, (445).