Artificial Intelligent Techniques with Watermarking

Nada N. Seleem **Baydaa I. Khaleel**
Nada_N_S@uomosul.edu.iq *baydaaibraheem@uomosul.edu.iq*
Shahbaa I. Khaleel
shahbaaibrkh@uomosul.edu.iq
College of Computer Sciences and Mathematics
University of Mosul

Received on: 18/08/2008 **Accepted on: 23/11/2008**

ABSTRACT

This research presents three robust blind watermarking algorithms in the discrete wavelet transform and spatial domain based on neural network and fuzzy logic artificial intelligent techniques. To enhance the performance of the watermarking system the first algorithm is developed by combining Radial Basis Function (*RBF)* neural network with Discrete Wavelet Transform (*DWT)* using *(DWT-RBFW)* algorithm for embedding and extracting of watermark. The second developed *(RBFW)* algorithm used *RBF* neural network for embedding and extracting of watermark based on intensity of whole image. The third developed *(FL-EXPW)* watermarking method is based on fuzzy logic and expert system techniques and it's the best algorithm among the three methods. The developed watermarking algorithms are robust against various attacks signal processing operations such as additive noise and jpeg compression, and geometric transformations.

Keyword: Watermarking system, Artificial intelligent techniques, Fuzzy logic, Artificial neural network.

استخدام تقنيات الذكاء الاصطناعي في العلامة المائية

ندى نعمت سليم **بيداء ابراهيم خليل** **شهباء ابراهيم خليل**

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2008/08/18 **تاريخ قبول البحث: 2008/11/23**

الملخص

قدم هذا البحث ثلاث خوارزميات كفوءة في موضوع العلامة المائية ذات الاستخلاص الأعمى التي لا تحتاج إلى وجود الصورة الأصلية باستخدام تحويل المويجة المتقطعة والمجال الحيزي بالاعتماد على تقنيات الذكاء الاصطناعي التي تتضمن الشبكات العصبية والمنطق المضبب. وقد انسئت الخوارزمية الاولى DWT-RBFW لتحسين أداء نظام العلامة المائية وذلك بدمج شبكة القاعدة الشعاعية (*RBF*) مع تحويل المويجة (DWT) لتضمين واستخلاص العلامة المائية. في الخوارزمية الثانية (*RBFW*) تم استخدام الشبكة العصبية (*RBF*) لتضمين العلامة المائية واستخلاصها بالاعتماد على شدة إضاءة الصورة بأكملها في المجال الحيزي. إنشئت

الخوارزمية الثالثة (*FL-EXPW*) بالاعتماد على تصميم النظام الخبير مع تقنية المنطق المضبب وكانت هذه الطريقة الأفضل بين الطرائق الثلاث. خوارزميات العلامة المائية الثلاث كانت كفوءة ضد الهجوم المتنوع لعمليات معالجة الإشارة مثل إضافة الضوضاء وإجراء عمليات الكبس والتحويلات الهندسية.

الكلمات المفتاحية: نظام العلامات المائية ، تقنيات ذكائية اصطناعية ، منطق ضبابي ، شبكة عصبية اصطناعية.

1. Introduction

With the rapid growth of the Internet and multimedia systems in distributed environments, it is easier for digital data owners to transfer multimedia documents across the Internet. Therefore, there is an increase in concern over copyright protection of digital contents [1,2]. Traditionally, encryption and control access techniques were employed to protect the ownership of media. These techniques, however, do not protect against unauthorized copying after the media have been successfully transmitted and decrypted. Recently, watermark techniques are utilized to maintain the copyright [2,3,4].

Unlike analog media such as audio and Video Home System (VHS) video tapes, multimedia data in digital form can be copied without degradation and freely distributed. Therefore, a major concern, with respect to protecting intellectual property rights, has arisen. One approach to address this problem is the embedding of an invisible digital watermark into multimedia data to mark the ownership. The embedded digital watermark may be copyright, authentication code, an imperceptible signature of the originator, or recipient of the host data [5].

The basic requirements of an effective robust watermarking scheme are:

- Imperceptibility: the watermark should be imperceptible when embedded in the host data. In other words, the watermark embedding process should not introduce any perceptible artifacts into the host data.
- Robustness: a watermark should remain intact in the host data regardless of any change that may occur to the host data, including all possible signals processing, and possible hostile attacks that unauthorized parties may attempt. Robustness against all possible attacks may be impossible to achieve. Thus the practical requirement is that the watermark is computationally unfeasible to remove without severely damaging the quality of the host signal [5].

Unfortunately, these two basic requirements are contradictory. For imperceptibility, the watermark embedding process should not introduce any perceptible artifacts into the host data. On the other hand, for high robustness, it is desirable that the watermark always involves a tradeoff between imperceptibility and robustness [5].

This research presented three methods for doing very robust and imperceptible watermarking using artificial neural network and fuzzy logic techniques.

2. previous works

Minfen Shen and Xinjung Zhang etc have proposed a method for digital image watermarking using ICA. This method is based on the independent component analysis for the digital image watermarking [2].

A new watermarking scheme using Backpropagation neural network with discrete wavelet transform is proposed by Maher Elarbi, etc. Which is presented in [13], and developed by this research.

New multi-Domains Image Watermarking Method Based on multi-Watermarks Embedding and Neural Network Segmentation was introduced in 2007 by Hassen Seddik,etc. This work proposes a method for image watermarking based on embedding identical watermarks in different domains of the image representation : spatial and DCT domains, without any distortion of the watermarked image [3].

A Blind Wavelet based image watermarking based on HVS and neural networks is presented by Hung-H. Tsai, chi-c. Liu and Kuo-C. Wang. This work proposes a blind wavelet based image watermarking technique, based on the human visible system model and neural network for image copyright protection [7].

3. Background

3.1 Digital Watermarking

There are two common approaches of performing watermarking: one in spatial domain, and the other in transformed domain. Each technique has its own advantages and disadvantages. In the spatial domain, the watermark is embedded into the host image by directly modifying the pixel value of the host image. The main advantage of the spatial domain watermarking schemes is that less computational cost is required. On the other hand, transformed domain watermarking schemes perform the domain transformation procedure by using transformation functions such as Discrete Wavelet Transformation (DWT), Discrete Cosine Transformation (DCT), Discrete Fourier Transformation (DFT), etc. Then, the transformed frequency coefficients are modified to embed the watermark bits. Finally, the inverse transformation function of the specific one used in the forward transformation procedure is performed. The main advantage of the frequency domain watermarking schemes is that they are more robust than the spatial domain schemes. However, they generally consume more computational cost because additional forward transformation and inverse transformation must be performed [6].

Figure (1) illustrates the general watermark embedding process. The inputs are the multimedia host data, the watermark and an optional public secret key. The watermark often consists of a binary data sequence, representing number, text, or even an image. The public or secret key is used to enforce security [5].

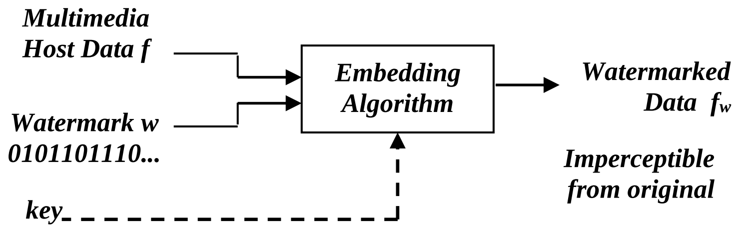


Fig. 1 : Watermark Embedding Process

The watermark sequence is embedded in the host data by making imperceptible modification to its content. The output of the watermark embedding is the modified, i.e. watermarked data [5].

The general watermark extraction process is depicted in figure (2), with the use of the key, an estimate of the original watermark is extracted from the watermarked multimedia data. For robust watermarking application, the watermark must be recoverable even when the watermarked data undergo a reasonable level of distortion [5].

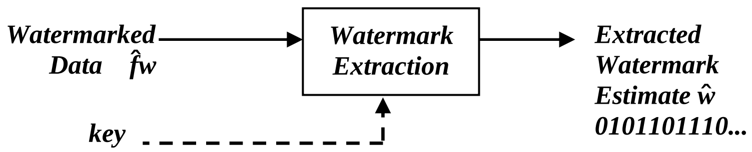
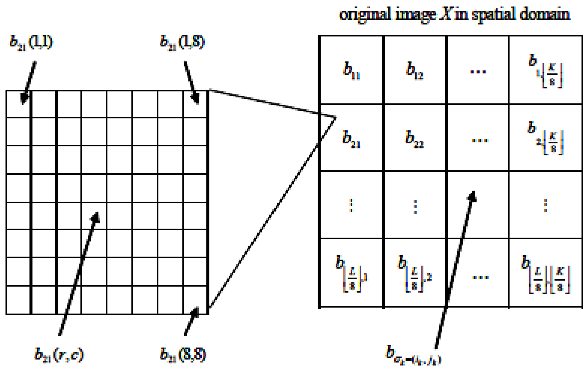


Fig. 2 : Watermark Extraction process

3.2. Image Denotations and DWT

A gray-level image, X, with size $L \times K$ can be defined by $X = [x_\rho]_{L\times K}$, where x_ρ Є{0, 1, …, 255}. Here x_ρ represents the pixel value located at position $\rho = (i, j)$ over X, where i Є {0, 1, …, L-1} and j Є {0, 1,…, K-1}. Figure (3) shows X is segmented into $\frac{L}{8}\times\frac{K}{8}$ non-overlapped blocks with size 8 × 8. For example, b_{21} stands for a block that the center pixel of the block is located at position (2, 1) on X. Let $b_{21}(r, c)$ denote the gray level of a pixel at the position (r, c) in the block b_{21}. As a result, these non-overlapped blocks

$\frac{L}{8}$ $\frac{K}{8}$



in X can be denoted by $\Phi = \{b_{ij} |\ i = 1, \ldots,\ \ , j = 1 \ldots,\ \ \}$ where b_{ij} stands for a size 8 × 8 block [7].

Fig. 3: The segmented image in spatial domain.

An image X is decomposed into LL, HL, LH and HH subbands through DWT transformation based on the linear-phase 9/7 wavelets. Figure (4) shows a block divided into 7 subbands with two levels DWT [7].

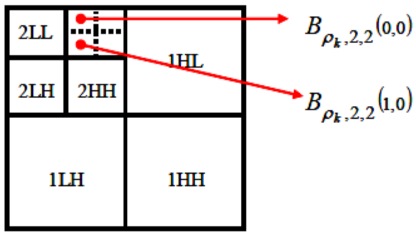


Fig 4: Components are constituted of a wavelet image block of a size 8 x 8 at level 2.

3.3 Soft Computing

Soft computing is a consortium of methodologies that works synergitically and provides in one form or another flexible information processing capability for handling real life ambiguous situations. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost by seeking for an approximate solution to an imprecisely/precisely formulated problem. Its major components, at this juncture, are fuzzy logic (FL), neural computing (NC), and probabilistic reasoning (PR), genetic algorithms (GA) and chaotic systems. FL is concerned mainly with providing algorithms for dealing with imprecision and approximate reasoning. NC provides the machinery for learning and curve fitting. PR, on the other hand, deals with probabilistic uncertainty, propagation of belief, and searching and optimization[8].

Genetic algorithms are randomized search and optimization techniques guided by the principles of evolution and natural genetics. They are efficient, adaptive, and robust search process, producing near-optimal solutions and having a large amount of implicit parallelism. Some applications of GAs in the areas of classification, pattern recognition, feature extraction, segmentation, image enhancement [9,10].

Artificial neural networks (ANNs) attempt to replicate the computational power of biological neural networks and, thereby, hopefully endow machines with some of the cognitive abilities that biological organisms possess. The various models are designated by the network topology, node characteristics, and the status updating rules. Tasks that ANNs can perform include pattern classification, clustering, optimization and control. ANNs can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edged are connections between neuron outputs and neuron inputs. On the basis of the connection pattern, ANNs can be grouped into two categories: Feedforward and Recurrent[11,12].

4. The Developed Watermarking System Techniques

We developed three watermark algorithms using fuzzy logic and radial basis function neural network artificial techniques as described in the following sections.

4.1 The *(DWT-RBFW)* Developed Method

In the first experiment, the watermark algorithm is performed in the 3-level *DWT* domain to an input original gray level and color image of size 256*256 pixel with a 2 bits watermark image of size 32*32 using *RBF* neural network.

During embedding and extracting of the watermark, three layers *RBF* network are used with 9, 4, 1 neurons in the input, hidden and output layer respectively. The input layer is simply a fan out layer and does no processing, the hidden layer performs a radial basis function. The final layer performs a simple weighted sum with linear output. The *RBF* network can establish the relationship between wavelet coefficients during training and memorize it between the original image and the watermarked image. The embedding and extracting are done by using *RBF* to train the neighbors of the selected coefficient according to the secret key, to insert value of watermark image instead of the value of selected coefficient according to the equations described in the algorithm.

4.1.1 Watermark embedding

The watermark embedding algorithm of the first technique is given as follows:

Step1. Input an original image I and watermark image *W*.

Step2. Compute and obtain the three levels *DWT* of the original image using haar filters.

Step3. According to the secret key a pseudo random coordinate sequence *index=(i,j)* is generated where *(i,j)Є [cv1,ch1,cd2,cv2,ch2,cd3 ,cv3,ch3]*.
And cv, ch, cd represents vertical, horizontal and diagonal coefficients, respectively of the three levels of discrete wavelet transform.

Step4. For a selected coefficient *I(i, j)* the Radial basis function network is trained with it's 3×3 neighbors as input vector and the value of coefficient *I(i, j)* as output values.

Step5. The watermark *W* is inserted by altering the value of the central coefficient of the original image according to the following formula:

$$Iw_{DWT}(i,j)=\begin{cases} max\{I_{DWT}(i,j),\sigma 1+\delta\} & if\ w_i=0 \\ min\{I_{DWT}(i,j),\sigma 0-\delta\} & if\ w_i=1 \end{cases} \quad \ldots(1)$$

Where w_i is the i^{th} bit of the watermark *W*, δ is the embedding strength, σ_1 and σ_0 are determined as follows:

$$\sigma 1=\begin{cases} RBF(i,j) & if(RBF(i,j)-I_{DWT}(i,j))>=\delta \\ I_{DWT}(i,j) & if(RBF(i,j)-I_{DWT}(i,j))<\delta \end{cases} \quad \ldots(2)$$

$$\sigma 0=\begin{cases} I_{DWT}(i,j) & if(RBF(i,j)-I_{DWT}(i,j))>=\delta \\ RBF(i,j) & if(RBF(i,j)-I_{DWT}(i,j))<\delta \end{cases} \quad \ldots(3)$$

Where *RBF(i, j)* is the output of the neural network.

Step6. Repeat steps 4-5 until all watermark bits are embedded into the original image *I* and obtain Iw_{DWT} image.

Step7. Output the watermarked image *Iw* by performed an inverse *DWT*.

When applying this method on color image the original image must be transformed from *RGB* color space to the *YCbCr* color space in step1 according to the following *formulas from reference [13,14]*:

$$Y = (77/256)R + (150/256)G + (29/256)B \quad \ldots(4)$$

$$Cb = -(44/256)R - (87/256)G + (131/256)B + 128 \quad \ldots(5)$$

$$Cr = (131/256)R - (110/256)G - (21/256)B + 128 \quad \ldots(6)$$

The eye is very sensitive to small changes in luminance, which is why only the luminance component *Y* undergoes watermark embedding. To obtain watermarked image *Iw* in step7 you must apply inverse *DWT* on Iw_{DWT} image followed by an inverse color space transformation according to the following formulas:

$$R = Y+1.371(Cr - 128) \quad \ldots(7)$$

$$G = Y - 0.698(Cr - 128) - 0.336(Cb - 128) \quad \ldots(8)$$

$$B = Y+1.732(Cb - 128) \quad \ldots(9)$$

4.1.2 Watermark extraction

The watermark extraction algorithm is described as follows:

Step1. Input a watermarked image *Iw* and if the original image is color then it must be transformed from *RGB* color space to the *YCbCr* color.

Step2. Compute and obtain the three levels *DWT*.

Step3. According to the secret key a pseudo random coordinate sequence *index = (i, j)* is generated where *(i, j) Є [cv1,ch1,cd2,cv2,ch2,cd3,cv3,ch3]*.

Step4. The watermark can be retrieved according to the Radial basis function network and the secret key as follows:

$$w = \begin{cases} 0 & if\ Iw_{DWT}(i,j) > RBF(i,j) \\ 1 & otherwise \end{cases} \quad \ldots(10)$$

Where Iw_{DWT} is the watermarked image and *RBF(i, j)* is the output of the neural network (Radial basis function neural network)

4.2 The *(RBFW)* Developed Method

In the second experiment, the watermark algorithm is performed to an input original gray level and color image of size 256*256 pixel with a 2 bits watermark image of size 32*32 using *RBF* neural network.

During embedding and extracting of the watermark, three layers *RBF* network are used with 8, 3, 1 neurons in the: input, hidden and output layer respectively, one neuron (bias neuron) was added in the input and hidden layer. The *RBF* network can establish the relationship between neighbor pixels during training and memorize it between the original image and the watermarked image. The embedding and extracting are done by using RBF neural network to train the neighbors of the selected pixel according to the secret key, of original image that used intensity of the whole image instead of some of *DWT* coefficients, to insert value of watermark image by altering the value of selected pixel according to the equations that described in the steps of algorithm.

4.2.1 Watermark embedding

The watermark embedding algorithm of the technique is given as follows:

Step1. Input an original image *I* and watermark image *W*.

Step2. According to the secret key a pseudo random coordinate sequence *index = (i, j)* is generated where *(i, j) Є I.*

Step3. For a selected pixel *I(i, j)* the Radial basis function network is trained with it's 3×3 neighbors as input vector and the value of pixel *I(i, j)* as output values.

Step4. The watermark *W* is inserted by altering the value of the central pixel of the original image according to the following formula:

$$Iw_{RBF}(i,j)=\begin{cases} max\{I_{Orginal}(i,j),\sigma 1+\delta\} & if\ w_i=0 \\ min\{I_{Orginal}(i,j),\sigma 0-\delta\} & if\ w_i=1 \end{cases} \quad \ldots(11)$$

Where w_i is the i^{th} bit of the watermark *W,* δ is the embedding strength, σ_1 and σ_0 are determined as follows:

$$\sigma 1=\begin{cases} RBF(i,j) & if(RBF(i,j)-I_{Orginal}(i,j))>=\delta \\ I_{Orginal}(i,j) & if(RBF(i,j)-I_{Orginal}(i,j))<\delta \end{cases} \quad \ldots(12)$$

$$\sigma 0=\begin{cases} I_{Orginal}(i,j) & if(RBF(i,j)-I_{Orginal}(i,j))>=\delta \\ RBF(i,j) & if(RBF(i,j)-I_{Orginal}(i,j))<\delta \end{cases} \quad \ldots\ldots(13)$$

Where *RBF(i, j)* is the output of the *RBF* neural network.

Step5. Repeat steps 3-4 until all watermark bits are embedded into the original image *I* and obtain Iw_{RBF} image.

When applying this method on color image the original image must be transformed from *RGB* color space to the *YCbCr* color space in step1 according to the equations (4-6). To obtain watermarked image Iw_{RBF} in step5 use inverse color space transformation equations (7-9).

4.2.2 Watermark extraction

The watermark extraction algorithm described as follows:

Step1. Input a watermarked image Iw_{RBF} and if the original image is color image transform it from *RGB* color space to the *YCbCr* color.

Step2. According to the secret key a pseudo random coordinate sequence *index = (i, j)* is generated where *(i, j) Є I.*

Step3. The watermark can be retrieved according to the Radial basis function network and the secret key as follows:

$$w=\begin{cases} 0 & if\ Iw_{RBF}(i,j)>RBF(i,j) \\ 1 & otherwise \end{cases} \quad \ldots(14)$$

Where Iw_{RBF} is the watermarked image and *RBF(i, j)* is the output of the neural network (Radial basis function neural network).

4.3 The *(FL-EXPW)* Developed Method

In the third experiment, the developed watermarking system is based on fuzzy logic and expert system techniques. The domain expert often has a good idea of what to expect in terms of input and output parameter values.

The expert can express these relationships in the form of rules. Fuzzy logic is used to handle the imprecision and vagueness of natural language and provides this additional advantages to a system.

Domain expert often express their knowledge in vague and imprecise linguistic terms. It is therefore a natural step to use the ability of fuzzy sets and fuzzy logic to model this imprecision.

The developed fuzzy expert system used fuzzy rules gained from a domain expert for watermark feature embedded.

The three inputs to the fuzzy inference system are the mean intensity of the pixel elements in three neighbors sub image. The output of the fuzzy inference system is used to embed the watermark. The three inputs and output composed of three membership functions shown in figure (5).

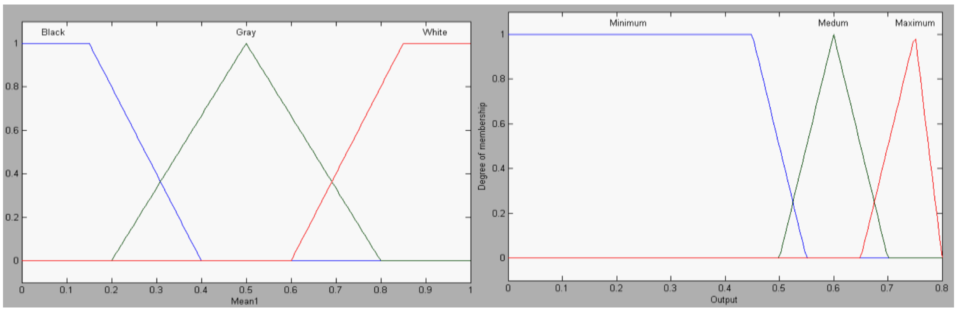


Figure (5): *FL-EXPW* input and output membership functions

The fuzzy expert system used total of 27 fuzzy rules provided in table (1). The fuzzy AND operation was performed using minimum function. Implication and aggregation were performed by minimum and maximum functions respectively. This method is better than other methods that are described in the previous sections.

Table (1): FL-EXPW watermarking fuzzy rules

Rule #	Mean-sub1	Mean-sub2	Mean-sub3	Output(alternative pixel value)
1	Black	Black	Black	Minimum
2	Black	Black	White	Medium
….	….	….	….	….
14	White	Gray	White	Maximum
15	White	Black	Gray	Medium
….	….	….	….	….
26	Gray	Black	Black	Minimum
27	Gray	White	White	Maximum

4.3.1 Watermark embedding and extraction

The Watermark embedding and extraction algorithm similar to the techniques used in section (4.2.1 and 4.2.2) with modification in step3 using fuzzy expert system instead of *RBF* neural network with three inputs sub images mean intensity.

The main steps of watermark embedding algorithm of this technique is given as follows:

Step1. Input an original image *I* and watermark image *W*.

Step2. According to the secret key a pseudo random coordinate sequence *index = (i, j)* is generated where *(i, j) Є I.*

Step3. For a selected pixel the three mean intensity are calculated of the three neighbors sub image with it's 3×3 size, These three parameters are inputs to the fuzzy inference system .

Step4. The watermark *W* is inserted by altering the value of the central coefficient of the original image *I(i, j)* according to the following formula:

$$Iw_{FLEXP}(i,j)=\begin{cases} max\{I_{Orginal}(i,j),\sigma 1+\delta\} & if\ w_i=0 \\ min\{I_{Orginal}(i,j),\sigma 0-\delta\} & if\ w_i=1 \end{cases} \quad \ldots(15)$$

Where w_i is the i^{th} bit of the watermark *W*, δ is the embedding strength, σ_1 and σ_0 are determined as follows:

$$\sigma 1=\begin{cases} FLEXP(i,j) & if(FLEXP(i,j)-I_{Orginal}(i,j))>=\delta \\ I_{Orginal}(i,j) & if(FLEXP(i,j)-I_{Orginal}(i,j))<\delta \end{cases} \quad \ldots(16)$$

$$\sigma 0=\begin{cases} I_{Orginal}(i,j) & if(FLEXP(i,j)-I_{Orginal}(i,j))>=\delta \\ FLEXP(i,j) & if(FLEXP(i,j)-I_{Orginal}(i,j))<\delta \end{cases} \quad \ldots(17)$$

Where *FLEXP(i, j)* is the output of the fuzzy inference system.

Step5. Repeat steps 3-4 until all watermark bits are embedded into the original image *I* and obtain Iw_{FLEXP} image.

When applying this method on color image the original image transformed from *RGB* color space to the *YCbCr* color space in step1 by using equations (4-6). To obtained watermarked image Iw_{FLEXP} in step4 you must apply an inverse color space transformation according to the equations (7-9).

The main steps of watermark extraction algorithm of this technique is given as follows:

Step1. Input a watermarked image Iw_{FLEXP} and if it is color image transform it from *RGB* color space to the *YCbCr* color.

Step2. According to the secret key a pseudo random coordinate sequence *index = (i, j)* is generated where *(i, j) Є I.*

Step3. The watermark can be retrieved according to the output of the fuzzy inference system and the secret key as follows:

$$w = \begin{cases} 0 & if\ \ Iw_{FLEXP}(i,j) > FLEXP(i,j) \\ 1 & otherwise \end{cases} \qquad \ldots(18)$$

Where Iw_{FLEXP} is the watermarked image and *FLEXP(i, j)* is the output of the fuzzy inference system.

5. Performance Measure

To demonstrate the performance of these methods, we used the standard gray level image "lena" and color image "peppers" with size of 256*256 pixels , to be watermarked, and test watermark image is a binary image with the size of 32*32 pixels, binary valued, as the watermark, see figure (6).

(a) (b) (c)

Figure (6): a) Original gray level image b) Original color image
c) Original watermark

The performance of the watermark extraction can be evaluated by estimating the normalized correlation coefficient for the extracted watermark and the original embedded watermark.

$$r = \frac{\sum_{i=1}^{M1}\sum_{j=1}^{M2} w(i,j)w^*(i,j)}{\sqrt{\sum_{i=1}^{M1}\sum_{j=1}^{M2} w(i,j)^2 \sum_{i=1}^{M1}\sum_{j=1}^{M2} w^*(i,j)^2}} \qquad \ldots(19)$$

Where both w and w^* represent the original and the extracted watermark, respectively, i and j are indexes of the binary watermark image. The value range of the normalized correlation coefficients is between minus one and unit. Obviously , the unity holds if the image is extracted perfectly matched the original. The minus sign indicates that the extracted image is a reverse version of its original image.[15,16]

The evaluation of the watermarked image quality is based on signal-to-noise-ratio *(SNR)* or Peak Signal to Noise Ratio *(PSNR)*, the definitions of *SNR* and *PSNR* are as follows:

$$SNR = 10 log_{10}\left(\frac{\sum_i Y_i^2}{\sum_i (Y_i - y_i)^2}\right) \quad \ldots(20)$$

Where Y and y are original image and watermarked image, respectively.

$$PSNR = 10. log_{10}\left(\frac{255^2}{MSE}\right) dB \quad \ldots(21)$$

$$MSE = \frac{1}{n.n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(I(i, j) - Iw(i, j))^2 \quad \ldots(22)$$

Where *MSE* is the mean square error, and $I(i, j)$ and $Iw(i, j)$ denotes the pixel values of the original image and the watermarked image.[15,17]

6. Experimental Results

In order to show the performance of the research watermarking methods, several experiments are carried out to compare between the three methods *DWT-RBFW* , *RBFW*, and *FL-EXPW* that are used in this work. Figure (7) shows the original gray level image, the watermark , the watermarked image, and the extracted watermark using *DWT-RBFW* method. Figure (8) shows the original gray level image, the watermark, the watermarked image , and the extracted watermark using *RBFW*. Figure (9) shows the original gray level image, the watermark, the watermarked image, and the extracted watermark using *FL-EXPW*. After calculating the *PSNR* of watermarked image by the three methods, the higher *PSNR* is obtained by using *FL-EXPW*. This ensures the ability of the third method to extract the watermarks perfectly in the absence of any attacks. We also notice that there is no difference between the original image and the watermarked image using human eye, this proved the fidelity of *FL-EXPW* method. The algorithms are coded in Matlab.

a) Original Image

b) Watermarked Image

c) Watermark

d) Extracted watermark

Figure (7) : a) Original Image b) Watermarked Image c) Watermark
d) Extracted watermark by using DWT-RBFW method

a) Original Image

b) Watermarked Image

c) Watermark

d) Extracted watermark

Figure (8) : a) Original Image b) Watermarked Image c) Watermark d) Extracted watermark by using RBFW method

a) Original Image

b) Watermarked Image

c) Watermark

d) Extracted watermark

Figure (9) : a) Original Image b) Watermarked Image c) Watermark d) Extracted watermark by using FL-EXPW method

Robustness against attacks is also a major watermarking requirement . various attacks were applied to the watermarking images to measure the robustness of the watermarking system. Three attacks strategies are used including rotation, translation, cropping (geometric transformations), additive noise two types of noise are added: Gaussian noise, salt and pepper noise, averaging filter, resizing, sharpening (signal processing) and damaging compression (such as jpeg compression).

To evaluate the robustness of *DWT-RBFW* method, *RBFW* method and *FL-EXPW* method, nine types of attacks were used to attack watermarked image.

Figure (10) shows the attacked watermarked images for *DWT-RBFW* method by 0.02 salt and pepper noise, Gaussian noise with zero mean and 0.01 variance, averaging filter, jpeg compression, resizing from size 256*256 to 128*128 and then to 256*256, cropping image, rotation by -15 degree, translation image i.e shifted down and right by 20 pixels, and by sharpening. Figure (11) show watermarks extracted by using *DWT-RBFW*. Table (2) shows the normalized correlation coefficient between each extracted watermark and the original watermark for the *DWT-RBFW* method. This method cannot extract complete watermark. In other words, the hidden watermark has been destroyed by the attacks.

a) attacked by 0.02 salt &pepper

b) attacked by 0 mean Gaussian noise with variance 0.01

c)attacked by averaging filter

d)attacked by jpeg compression

e)attacked by resizing from 256*256 -128*128-256*256

f) attacked by cropping

g) attacked by rotating -15 degree

h) attacked by translate down & right by 20 pixels

i) attacked by sharpening

Figure (10) : Watermarked images after attacked by nine types of attacks using DWT-RBFW method

a) 0.02 salt &pepper

b) Gaussian noise with 0 mean & variance 0.01

c) averaging filter

d) jpeg compression

e) resizing 256 -128-256

f) cropping

g) rotating -15 degree h) translate by 20 pixels i) sharpening

Figure (11) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *DWT-RBFW* method

Table (2): The Normalized Correlation Coefficient and PSNR between each extracted watermark and the original watermark for the DWT-RBFW method

Attack	*Normalized Correlation Coefficient*	*PSNR*
salt & pepper noise	0.8235	53.5503
Gaussian noise	0.8280	53.3483
averaging filter	0.7070	53.0618
jpeg compression	0.8189	53.4916
Resizing	0.8047	53.2095
cropping	0.6584	51.1073
rotating	0.6717	51.2354
Translate	0.6784	51.3230
sharpening	0.8246	53.6248

Figure (12) shows the attacked watermarked images for *RBFW* method after applying the same attacks we applied on *DWT-RBFW* method. Figure (13) shows the extracted watermarks from attacked watermarked images, and table (3) shows the normalized correlation coefficient between each extracted watermark and the original watermark.

a) attacked by 0.02 salt &pepper b) attacked by 0 mean Gaussian noise with c) attacked by averaging filter variance 0.01

d) attacked by jpeg compression

e) attacked by resizing from 256*256 - 128*128-256*256

f) attacked by cropping

g) attacked by rotating - 15 degree

h) attacked by translate down & right by 20 pixels

i) attacked by sharpening

Figure (12) : Watermarked images after attacked by nine types of attacks using RBFW method

a) 0.02 salt &pepper

b) Gaussian noise with 0 mean & variance 0.01

c) averaging filter

d) jpeg compression

e) resizing 256 -128-256

f) cropping

g) rotating -15 degree   h) translate by 20 pixels   i) sharpening

Figure (13) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *RBFW* method

Table (3): The Normalized Correlation Coefficient and PSNR between each extracted watermark and the original watermark for the RBFW method

Attack	***Normalized Correlation Coefficient***	***PSNR***
salt & pepper noise	0.8386	53.9687
Gaussian noise	0.8315	53.8563
averaging filter	0.8104	53.3342
jpeg compression	0.9315	57.4783
Resizing	0.8585	54.4863
cropping	0.8249	53.4337
rotating	0.6865	51.3853
Translate	0.7230	51.8191
sharpening	0.9272	57.2301

Figure (14) shows the attacked watermarked images for *FL-EXPW* method after applying the same attacks we applied on *DWT-RBFW* and *RBFW* methods. Figure (15) shows the extracted watermarks from attacked watermarked images, and table (4) shows the normalized correlation coefficient between each extracted watermark and the original watermark. All the watermarks were completely extracted by this method. Therefore, this method has high fidelity and robustness against geometric attacks and signal processing operation such as additive noise and jpeg compression.

a)attacked by 0.02 salt &pepper

b) attacked by 0 mean Gaussian noise with variance 0.01

c)attacked by averaging filter

d) attacked by jpeg compression

e) attacked by resizing from 256*256 -128*128-256*256

f) attacked by cropping

g) attacked by rotating -15 degree

h) attacked by translate down & right by 20 pixels

i) attacked by sharpening

Figure (14) : Watermarked images after attacked by nine types of attacks using *FL-EXPW* method

a) 0.02 salt &pepper

b) Gaussian noise with 0

c) averaging filter

mean & variance 0.01

d) jpeg compression e) resizing 256 -128-256 f) cropping

g) rotating -15 degree h) translate by 20 pixels i) sharpening

Figure (15) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *FL-EXPW* method

Table (4): The Normalized Correlation Coefficient and PSNR between each extracted watermark and the original watermark for the FL-EXPW method

Attack	***Normalized Correlation Coefficient***	***PSNR***
salt & pepper noise	0.8758	55.0323
Gaussian noise	0.8578	54.3956
averaging filter	0.8493	54.2544
jpeg compression	0.9426	58.2338
Resizing	0.8630	54.6354
cropping	0.9532	59.0957
rotating	0.7873	52.9063
Translate	0.8226	53.6098
sharpening	0.9938	67.8199

In addition to all experiments above that proved the *FL-EXPW* is the best method , we applied the three methods on color image. Figure (16) shows the original color image, the watermark , the watermarked image, and the extracted watermark using *DWT-RBFW* method. After calculating the *PSNR* of watermarked image by the three methods, see table (5), the higher *PSNR* is obtained by using *FL-EXPW*. This ensures the ability of the third method to extract the watermarks perfectly in the absence of any attacks. We also notice that there is no difference between the original image and the watermarked image using human eye, this proved the fidelity of *FL-EXPW* method.

Table (5): The PSNR between each watermarked and the original image for the three methods that are used in this research

Methods	***DWT-RBFW***	***RBFW***	***FL-EXPW***
PSNR	35.3112	38.2337	42.2346
SNR	29.3339	32.2564	36.2573

a) Original Color Image   b) Watermarked Image

c) Watermark

d) Extracted watermark

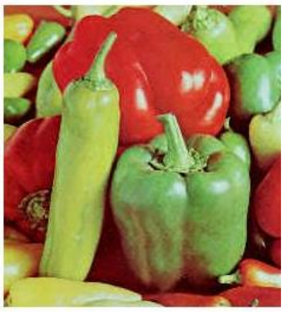
Figure (16) : a) Original Color Image b) Watermarked Image c)Watermark d) Extracted watermark by using DWT-RBFW method

To evaluate the robustness of *DWT-RBFW* method, *RBFW* method and *FL-EXPW* method, nine types of attacks were used to attack watermarked image. Figure (17) shows the attacked watermarked images for *DWT-RBFW* method after applying the same attacks we applied on gray level image "lena". Figure (18, 19, 20) show the extracted watermarks from attacked watermarked images, and table (6, 7, 8) shows the normalized correlation coefficient between each extracted watermark and the original watermark by using *DWT-RBFW*, *RBFW*, and *FL-EXPW* method, respectively. From the tables high normalized correlation coefficients are obtained when using *FL-EXPW* method, which proves that the embedded watermarks were not damaged by the applied nine attacks. Therefore the *FL-EXPW* is considered the best method. Appendix (A) shows the results of the three methods on gray level and color image after applying nine attacks on it.

a)attacked by 0.02 salt &pepper

b) attacked by 0 mean Gaussian noise with variance 0.01

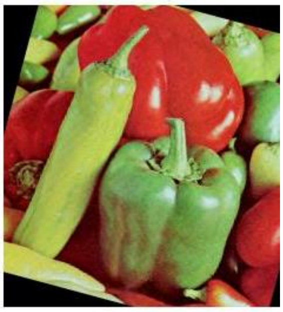
c)attacked by averaging filter



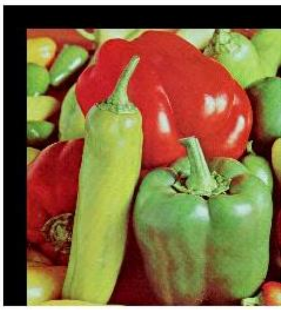
d)attacked by jpeg compression

e)attacked by resizing from 256*256 -128*128-256*256

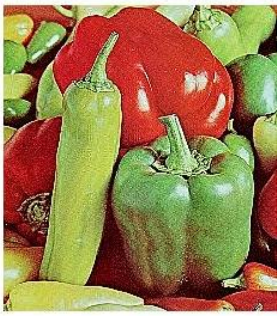
f) attacked by cropping



g) attacked by rotating -15 degree



h) attacked by translate down & right by 20 pixels



i) attacked by sharpening

Figure (17) : Watermarked images after attacked by nine types of attacks using DWT-RBFW method

a) 0.02 salt &pepper

b) Gaussian noise with 0 mean & variance 0.01

c) averaging filter

d) jpeg compression

e) resizing 256 -128-256

f) cropping

g) rotating -15 degree

h) translate by 20 pixels

i) sharpening

Figure (18) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *DWT-RBFW* method

each extracted watermark and the original watermark for the DWT-RBFW method

Attack	***Normalized Correlation Coefficient***	***PSNR***
Salt & pepper noise	0.8394	53.9850
Gaussian noise	0.8314	53.7934
averaging filter	0.8044	53.1959
jpeg compression	0.8395	53.9850
Resizing	0.8327	53.8090
cropping	0.7030	51.6062
rotating	0.6710	51.2354
Translate	0.6914	51.4669
sharpening	0.8291	53.7313

a) 0.02 salt &pepper

b) Gaussian noise with 0 mean & variance 0.01

c) averaging filter

d) jpeg compression

e) resizing 256 -128-256

f) cropping

g) rotating -15 degree

h) translate by 20 pixels

i) sharpening

Figure (19) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *RBFW* method

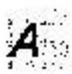
Table (7): The Normalized Correlation Coefficient and PSNR between each extracted watermark and the original watermark for the RBFW method

Attack	***Normalized Correlation Coefficient***	***PSNR***
salt & pepper noise	0.8661	54.7120
Gaussian noise	0.8536	54.3421
averaging filter	0.8080	53.2784
jpeg compression	0.9207	56.8666
Resizing	0.8751	54.9910
cropping	0.8507	54.0508
rotating	0.6794	51.3230
Translate	0.7767	52.6347
sharpening	0.9391	57.9807

a) 0.02 salt &pepper

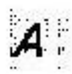
b) Gaussian noise with 0 mean & variance 0.01

c) averaging filter

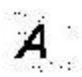


d) jpeg compression

e) resizing 256 -128-256



f) cropping



g) rotating -15 degree h) translate by 20 pixels i) sharpening

Figure (20) : The extracted watermarks of Watermarked images after attacked by nine types of attacks using *FL-EXPW* method

Table (8): The Normalized Correlation Coefficient and PSNR between each extracted watermark and the original watermark for the FL-EXPW method

Attack	*Normalized Correlation Coefficient*	*PSNR*
salt & pepper noise	0.8978	55.8283
Gaussian noise	0.9052	56.1387
averaging filter	0.8959	55.7541
jpeg compression	0.9671	60.5995
Resizing	0.9132	56.5019
cropping	0.9734	61.5128
rotating	0.8513	54.3068
Translate	0.9003	55.9293
sharpening	0.9899	65.6811

7. Conclusions

This research has discussed the watermark system and the enhanced methods for digital image watermark processing using artificial intelligent techniques. In addition, the extraction process can be performed blindly. It provides successful retrieval of the watermark image not only from the watermarked image but also from the corrupted watermarked image. It has been observed that recovery of the watermark image is possible even if the watermarked image is corrupted by 0.02 salt and pepper noise, Gaussian noise with zero mean and 0.01 variance, averaging filter, jpeg compression, resizing from size 256*256 to 128*128 and then to 256*256, cropping image, rotation by -15 degree, translation image i.e shifted down and right by 20 pixels, and by sharpening. In this research we developed three enhanced methods *DWT-RBFW*, *RBFW* and *FL-EXPW*. The performance analysis was carried by estimating the normalized correlation coefficient and peak signal to noise ratio which showed that the (*RBFW*) gives results better than the (*DWT-RBFW*) method, and the third method (*FL-EXPW*) which used fuzzy logic and expert system gives results much better than other methods to extract watermarks even in the presence of nine attacks. Experimental results demonstrate the robustness and fidelity of the third method(*FL-EXPW*).

REFERENCES

[1] A. Piva, F. Bartolini, and M. Barni, "Managing copyright in open networks", IEEE Transactions on Internet Computing, Vol. 6, Issue.3, pp. 18-26, May-June 2002.

[2] J. Lee and S. Jung, "A survey of watermarking techniques applied to multimedia", Proceedings 2001 IEEE International Symposium on Industrial Electronics(ISIE2001), Vol. 1, pp. 272-277, 2001.

[3] M. Barni, F. Bartolini, R. Caldelli, A. De Rosa, and A. Piva, "A Robust Watermarking Approach for Raw Video, "Proceedings 10th International Packet Video Workshop PV2000, Cagliari, Italy, May 1-2 2000.

[4] A. Eskicioglu and E. Delp, "An overview of multimedia content protection in consumer electronics devices", Proceedings Signal Processing Image Communication 16 (2001), pp. 681-699.

[5] Chuhong Fei, "On the choice of transform for robust watermarking in the presence of lossy compression", Master thesis (2001).

[6] Shwu-Huey Yen and Chia-Jen Wang, "SVM Based Watermarking Technique", Tamkang Journal of Science and Engineering, Vol. 9, No 2, pp. 141_150 (2006).

[7] Hung H. Tsai, Chi C. Liu and Kuo C. Wang, “Blind Wavelet-based Image Watermarking Based on HVS and Neural Networks“, www.**atlantis-press.com**/php/paper-details.php?id=40

[8] S. K. Pal, N. R. Pal, "Soft computing: Goals, tools and feasibility", Jornal of the Institute of Elecronics and Telecommunications Engineers, vol. 42, pp. 195-204, 1997.

[9] S. K. Pal, S. Bandyopadhyay and C, A. Murthy, "Genetic algorithms for generation of class boundaries", IEEE Transactions on systems, Man, and Cybernetics, Part B, vol. 25, pp.816-828, 1998.

[10] S. Bandyopadhyay, S. K. Pal and U. Maulik, "Incorporating chromosome differentiation in genetic algorithms", Information Sciences, vol. 104, pp. 293-319, 1998.

[11] A. K. Jain and J. Mao, "Artificial neural networks: A tutorial", IEEE Computer, March 1996.

[12] S. K. Pal and P. K. Srimani, eds., "Special Issue on Neural Networks: Theory and Applications", IEEE Computer, vol. 29, no. 3. 1996.

[13] Maher ELARBI, Chokri BEN AMAR and Adel ALIMI, “A novel approach using neural network with wavelet coefficients for image watermarking”. **www.regim.org/publications/conferences/2005/2005_ISCIII_EBA.pdf**

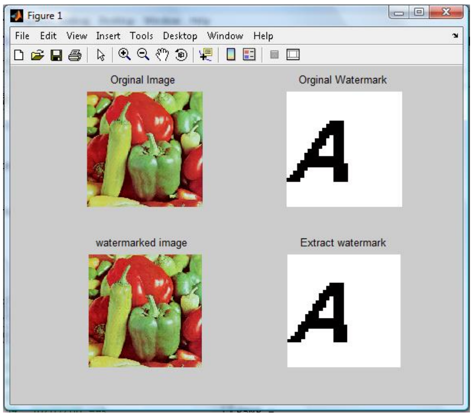
[14] David Salomon, “Data Compression” third addition, Springer, 2004.

[15] M. Ketcham, and S. Vongpradhip, “Intelligent Audio Watermarking using Genetic Algorithm in DWT Domain”, International Journal of Intelligent technology, vol. 2, no. 2, pp. 135-140,2007.

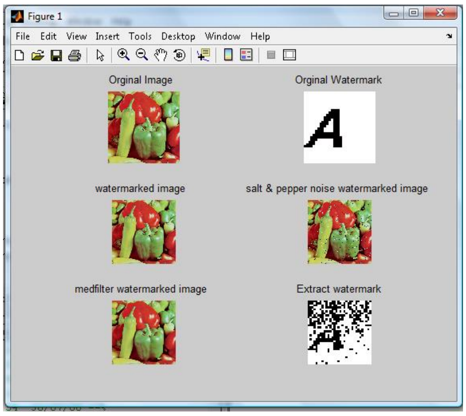
[16] M. Shen, X. Zhang, L. Sun, P. Beadle and F. Chan, “A method for digital image watermarking using ICA”,4th International symposium on independent component analysis and blind signal separation(ICA2003),2003,Nara, Japan.

[17] H. Seddik, M. Sayadi, F. Fnaiech, and M. Cheriet, “New multi-domains image watermarking method based on multi-watermarks embedding and neural network segmentation, Applied mathematical sciences, vol. 1, no. 39, pp. 1927-1939,2007.

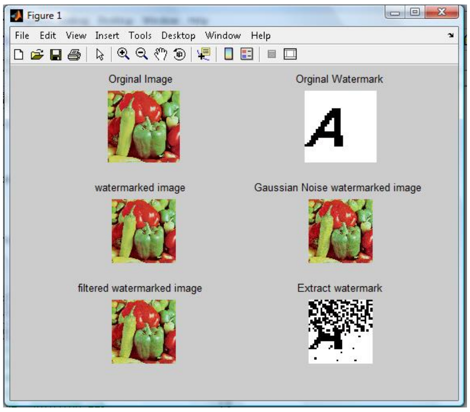
Appendix A



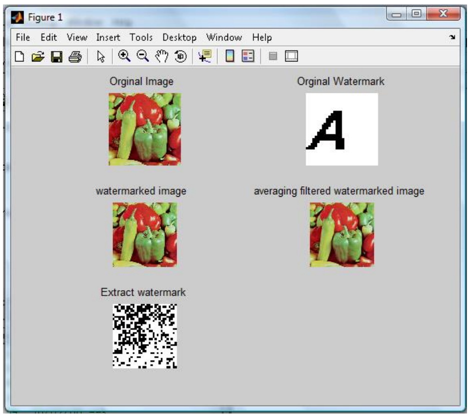
With out any attack using DWT-RBFW method



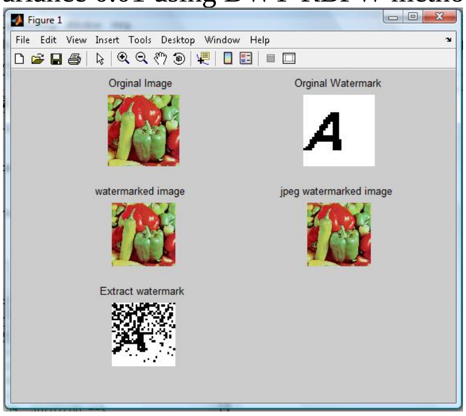
attacked by 0.02 salt &pepper using DWT-RBFW method



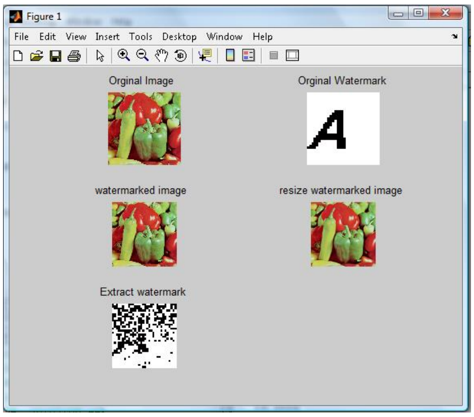
attacked by 0 mean Gaussian noise with variance 0.01 using DWT-RBFW method



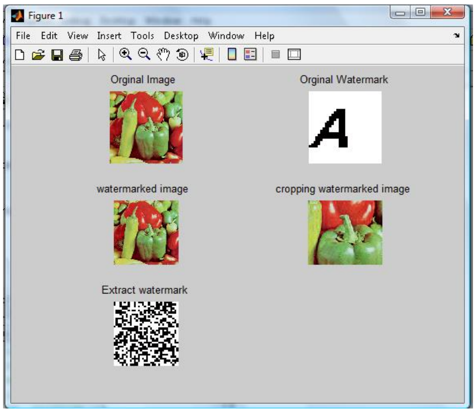
attacked by averaging filter using DWT-RBFW method



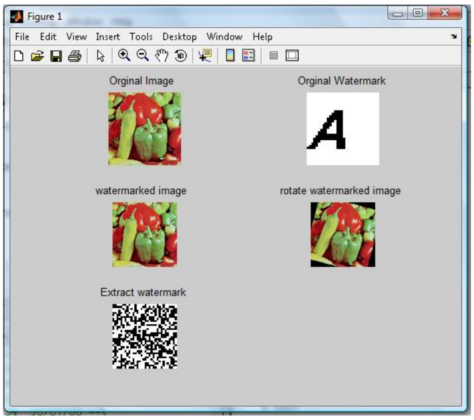
attacked by jpeg compression using DWT-RBFW method



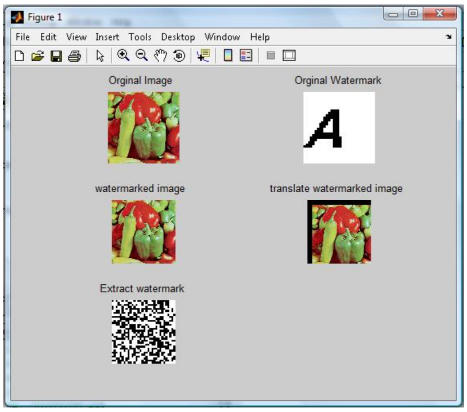
attacked by resizing from 256*256 - 128*128-256*256 using DWT-RBFW method



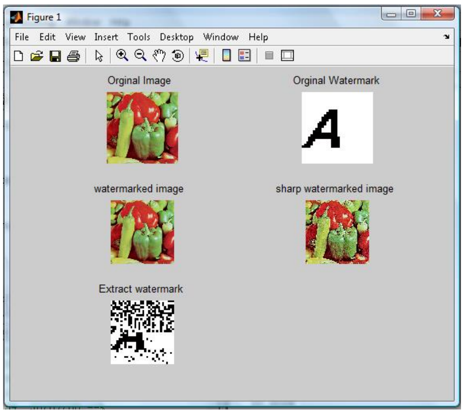
attacked by cropping using DWT-RBFW method



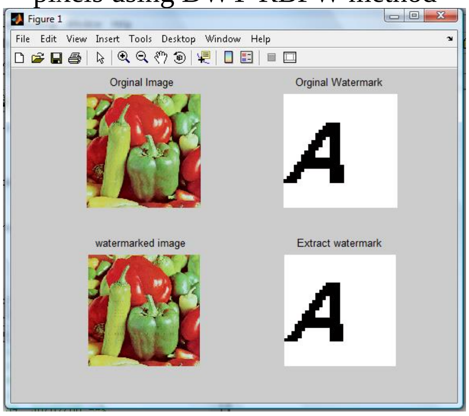
attacked by rotating -15 degree using DWT-RBFW method



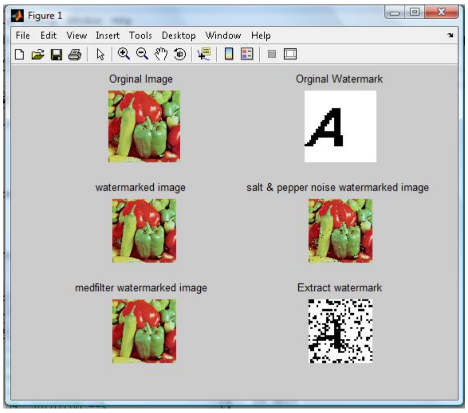
attacked by translate down & right by 20 pixels using DWT-RBFW method



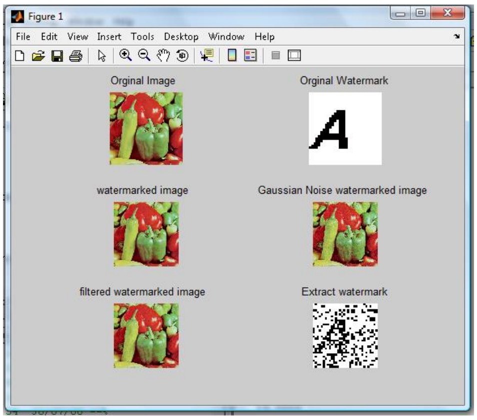
attacked by sharpening using DWT-RBFW method



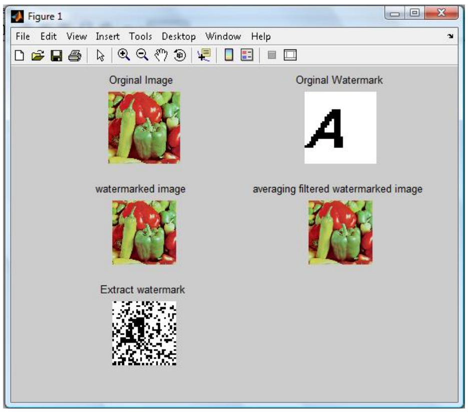
With out any attack using RBFW method



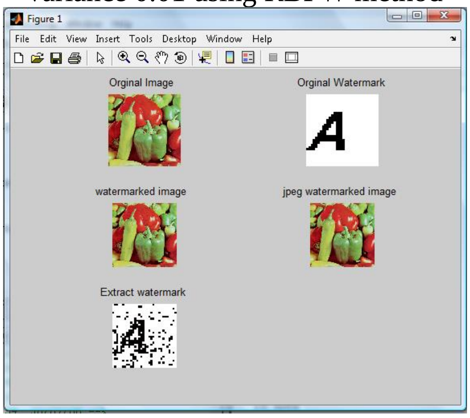
attacked by 0.02 salt &pepper using RBFW method



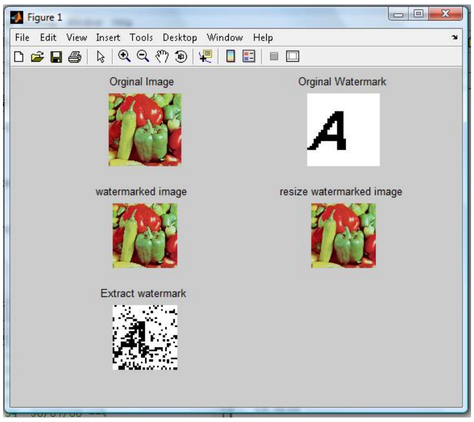
attacked by 0 mean Gaussian noise with variance 0.01 using RBFW method



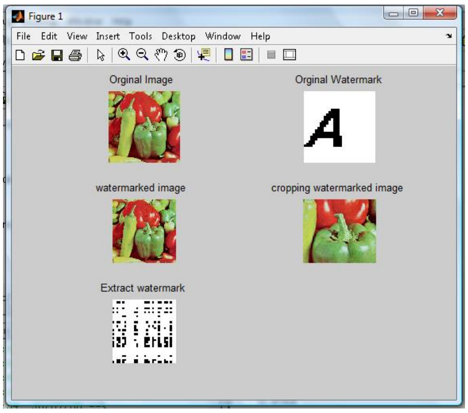
attacked by averaging filter using RBFW method



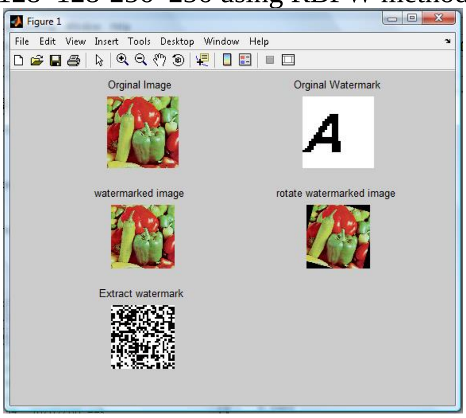
attacked by jpeg compression using RBFW method



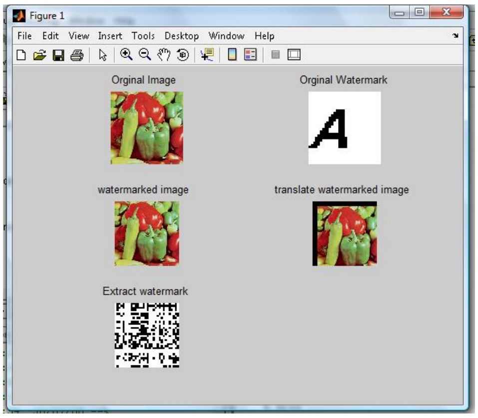
attacked by resizing from 256*256 - 128*128-256*256 using RBFW method



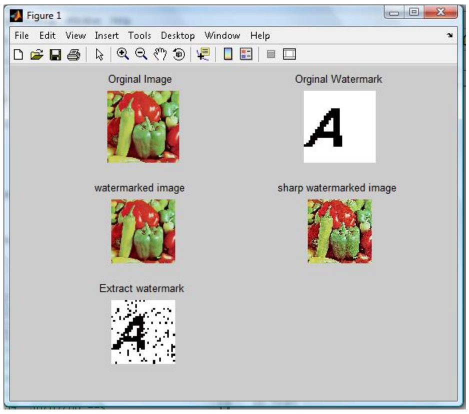
attacked by cropping using RBFW method



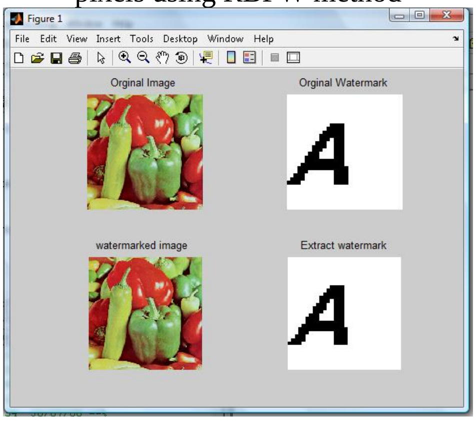
attacked by rotating -15 degree using RBFW method



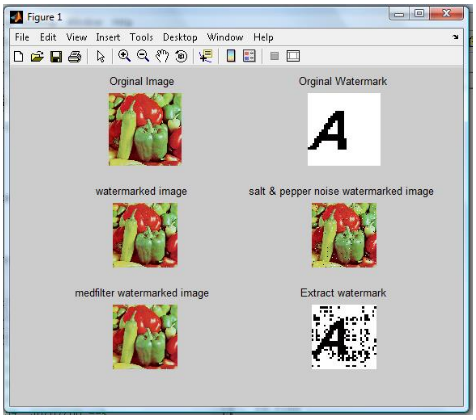
attacked by translate down & right by 20 pixels using RBFW method



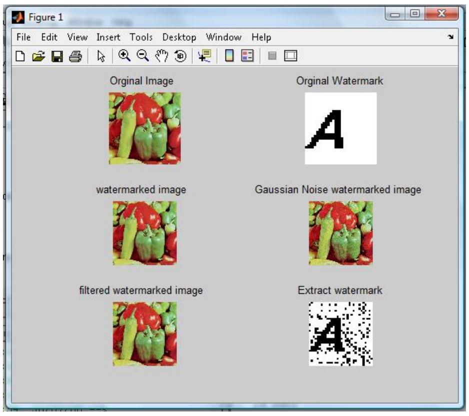
attacked by sharpening using RBFW method



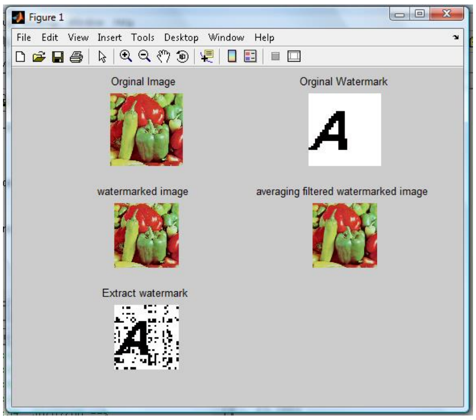
With out any attack using FL-EXPW method



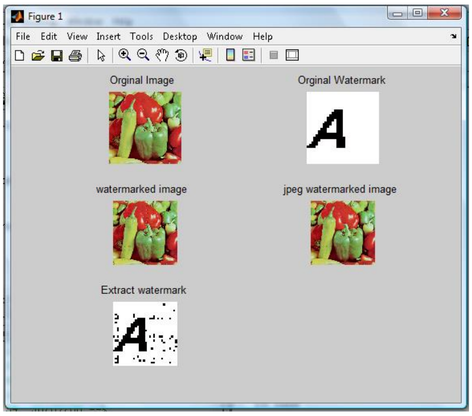
attacked by 0.02 salt &pepper using FL-EXPW method



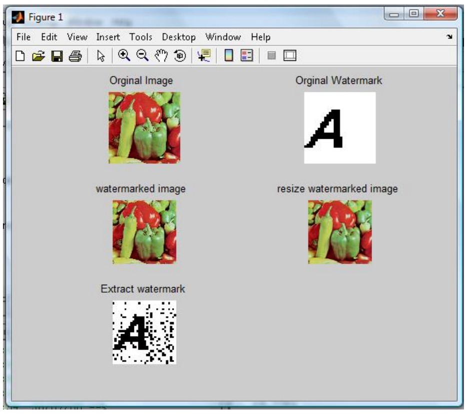
attacked by 0 mean Gaussian noise with variance 0.01 using FL-EXPW method



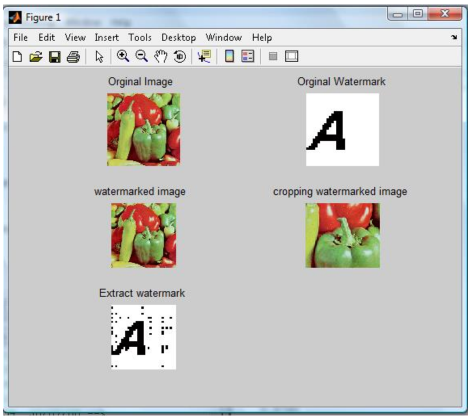
attacked by averaging filter using FL-EXPW method



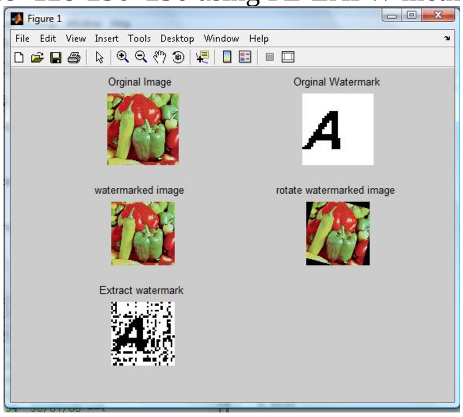
attacked by jpeg compression using FL-EXPW method



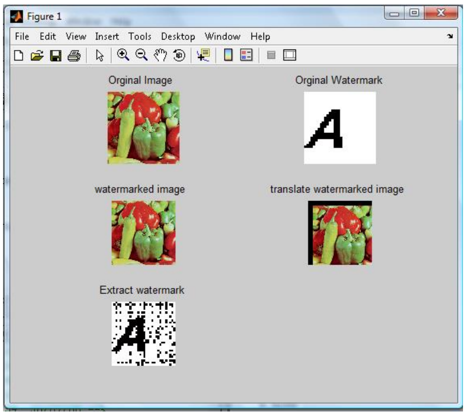
attacked by resizing from 256*256 -128*128-256*256 using FL-EXPW method



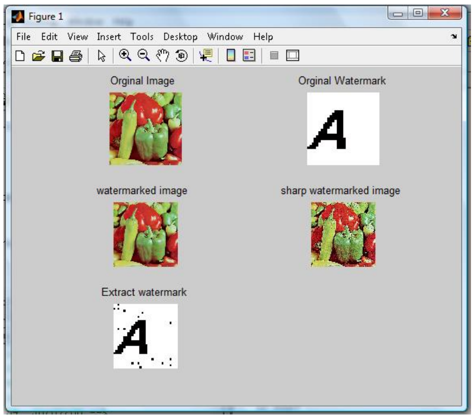
attacked by cropping using FL-EXPW method



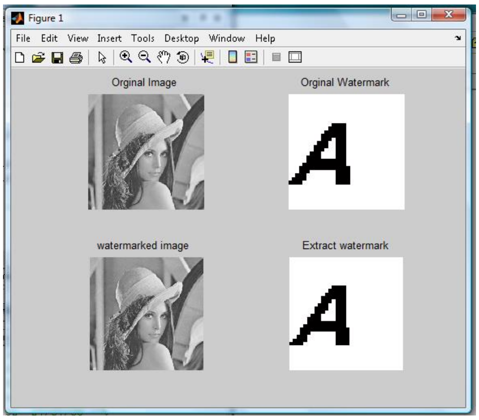
attacked by rotating -15 degree using FL-EXPW method



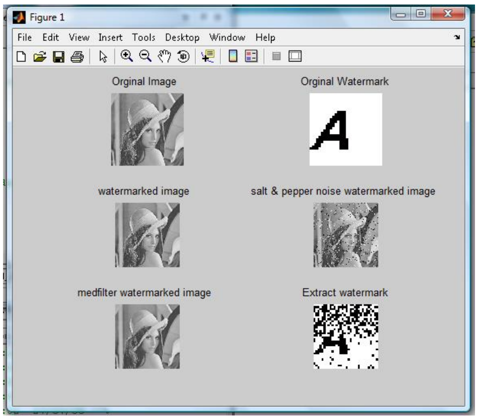
attacked by translate down & right by 20 pixels using FL-EXPW method



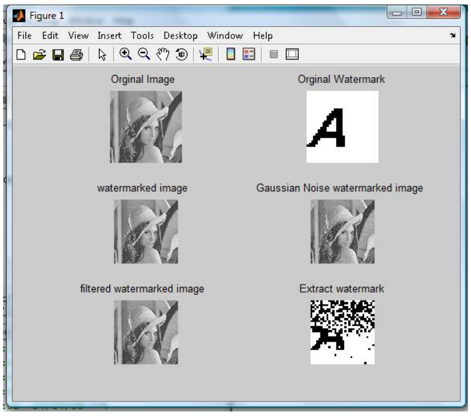
attacked by sharpening using FL-EXPW method



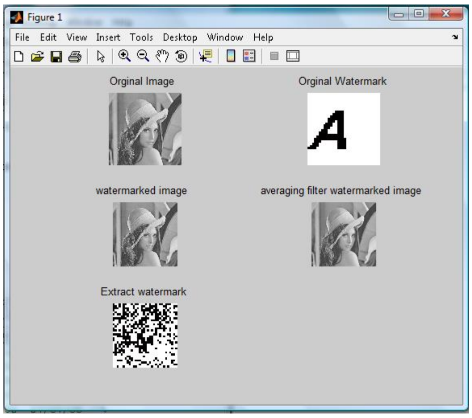
With out any attack using DWT-RBFW method



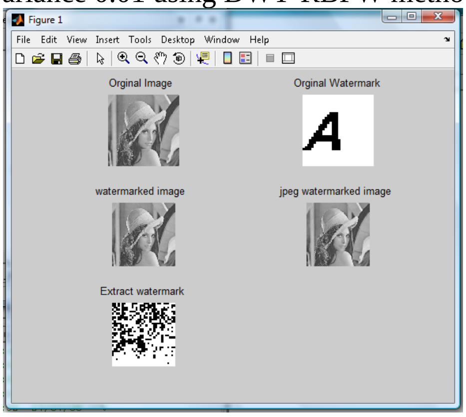
attacked by 0.02 salt &pepper using DWT-RBFW method



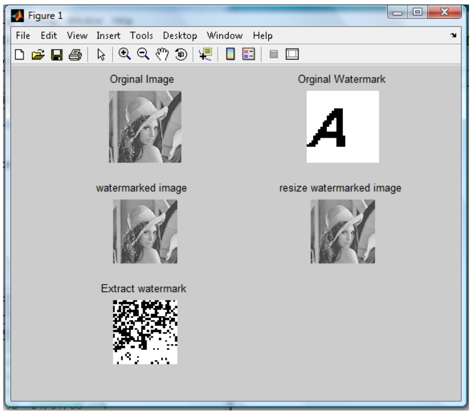
attacked by 0 mean Gaussian noise with variance 0.01 using DWT-RBFW method



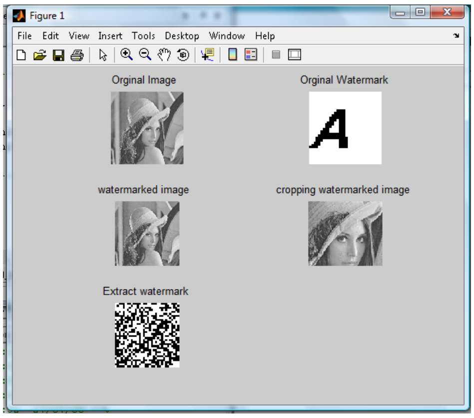
attacked by averaging filter using DWT-RBFW method



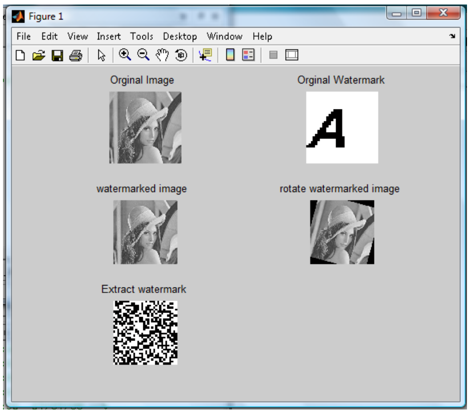
attacked by jpeg compression using DWT-RBFW method



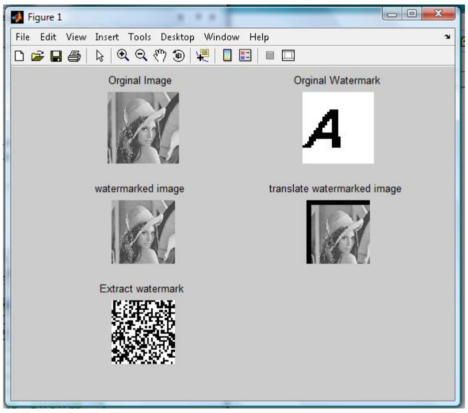
attacked by resizing from 256*256 -128*128-256*256 using DWT-RBFW method



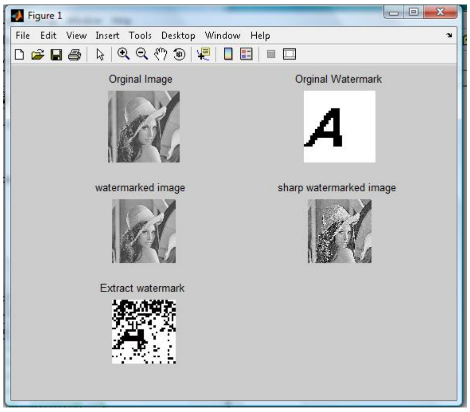
attacked by cropping using DWT-RBFW method



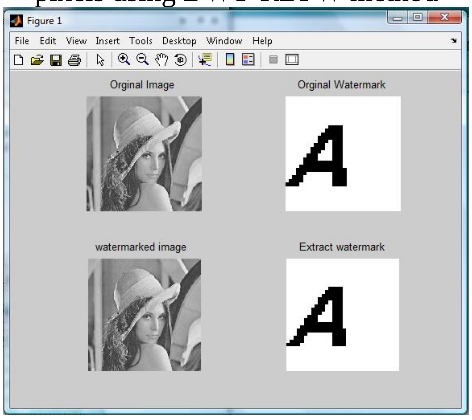
attacked by rotating -15 degree using DWT-RBFW method



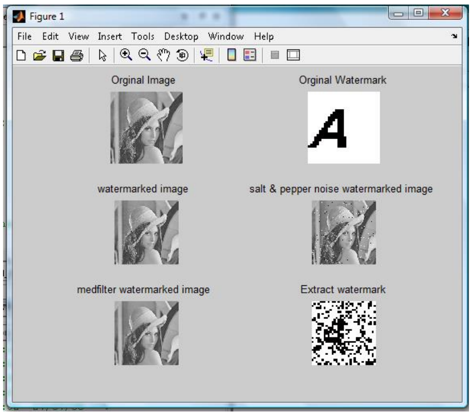
attacked by translate down & right by 20 pixels using DWT-RBFW method



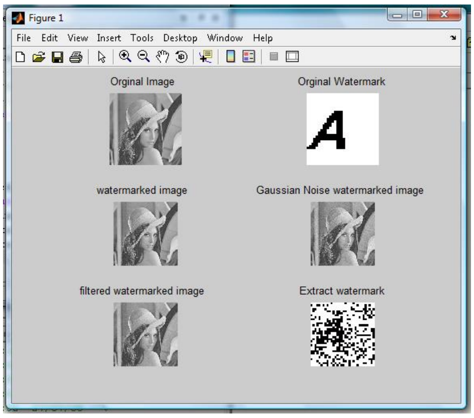
attacked by sharpening using DWT-RBFW method



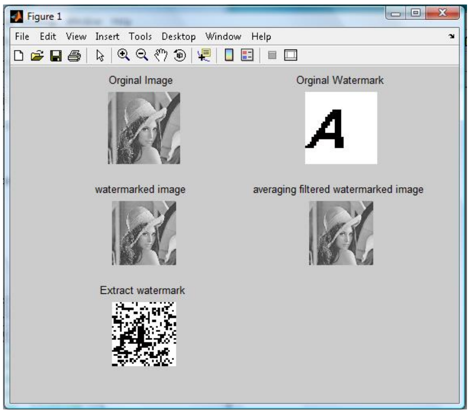
With out any attack using RBFW method



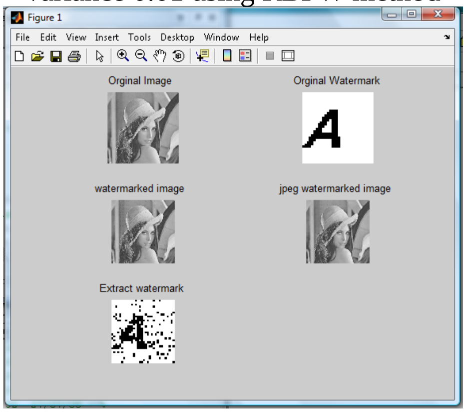
attacked by 0.02 salt &pepper using RBFW method



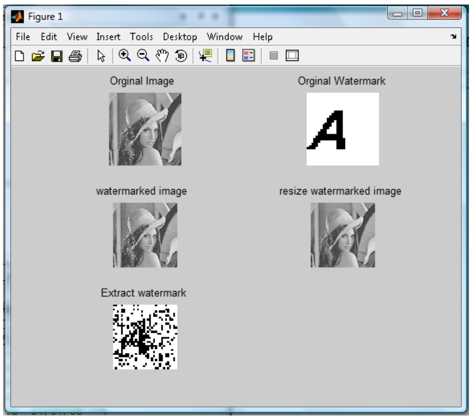
attacked by 0 mean Gaussian noise with variance 0.01 using RBFW method



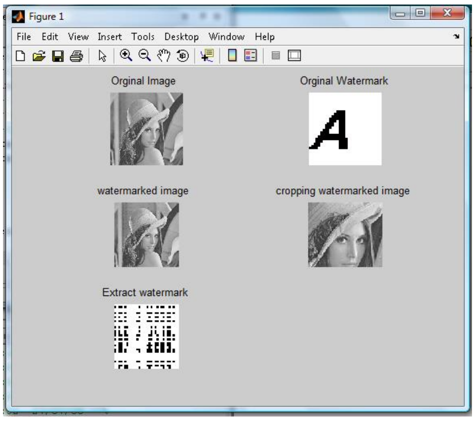
attacked by averaging filter using RBFW method



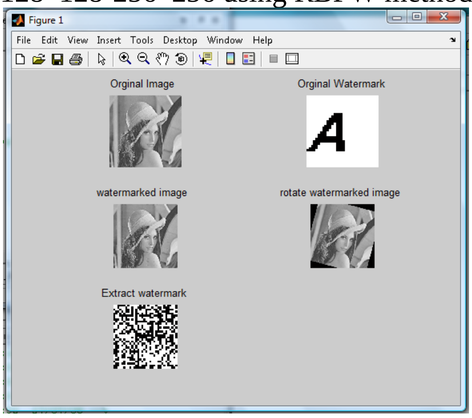
attacked by jpeg compression using RBFW method



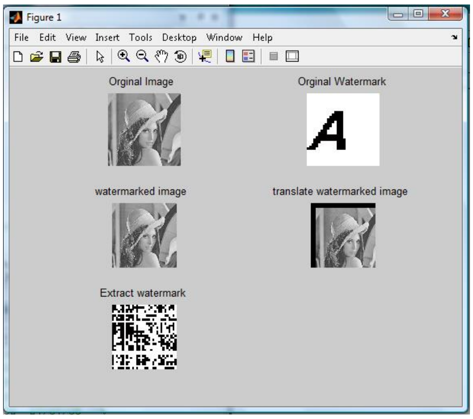
attacked by resizing from 256*256 -128*128-256*256 using RBFW method



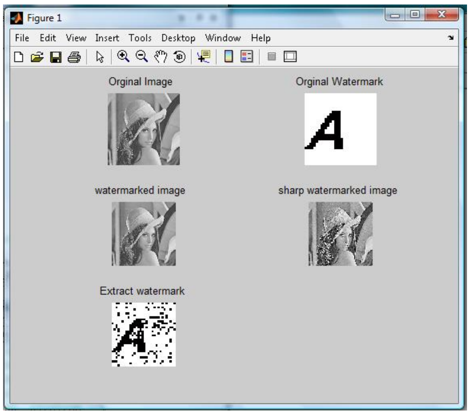
attacked by cropping using RBFW method



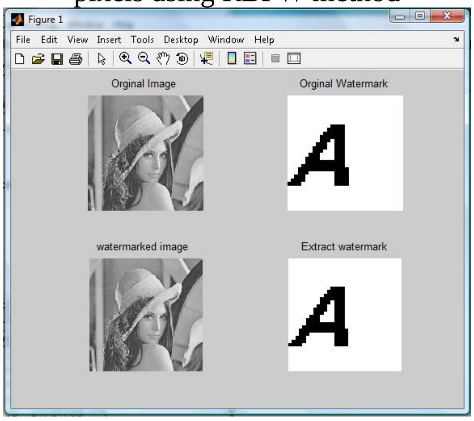
attacked by rotating -15 degree using RBFW method



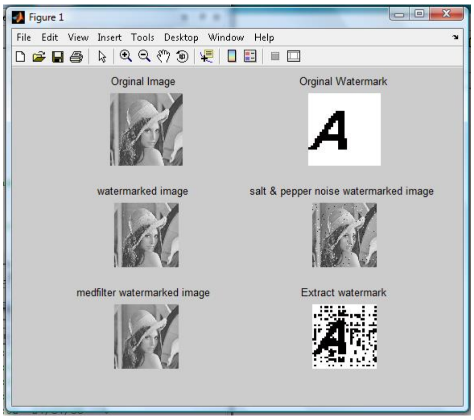
attacked by translate down & right by 20 pixels using RBFW method



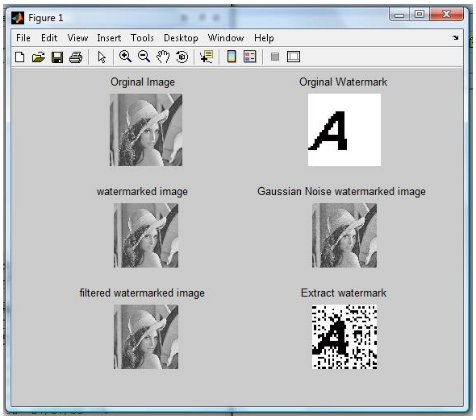
attacked by sharpening using RBFW method



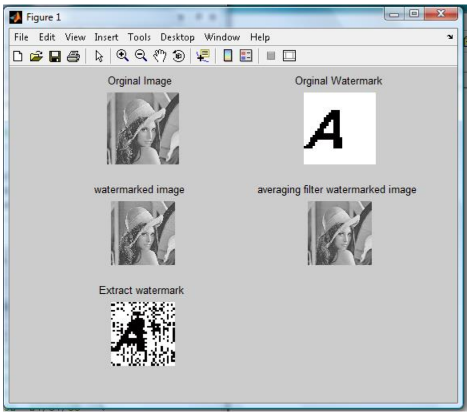
With out any attack using FL-EXPW method



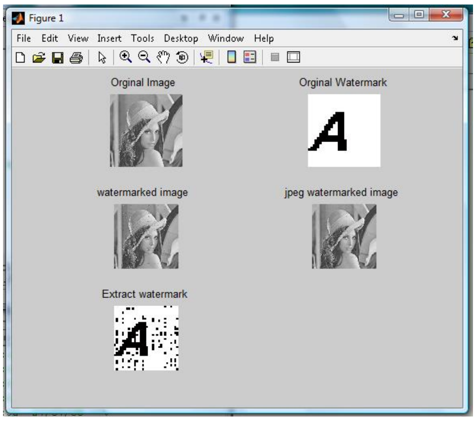
attacked by 0.02 salt &pepper using FL-EXPW method



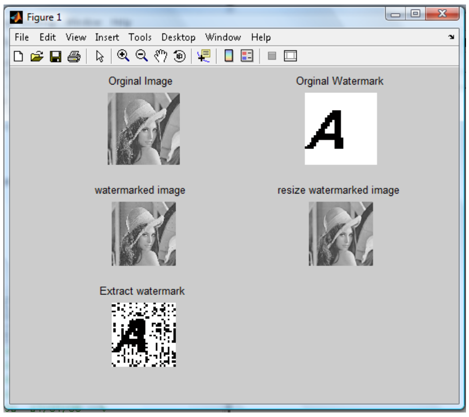
Attacked by 0 mean Gaussian noise with variance 0.01 using FL-EXPW method



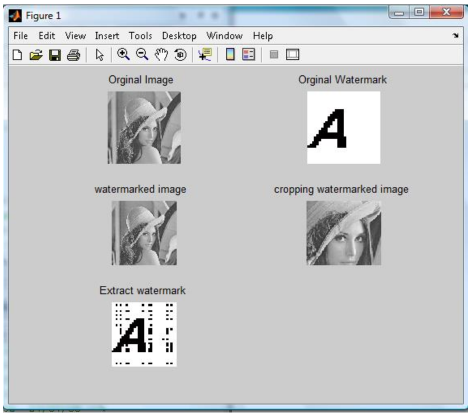
attacked by averaging filter using FL-EXPW method



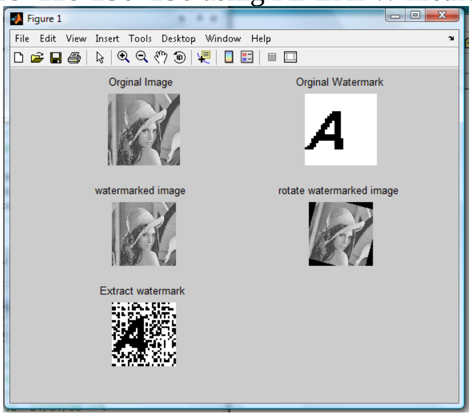
Attacked by jpeg compression using FL-EXPW method



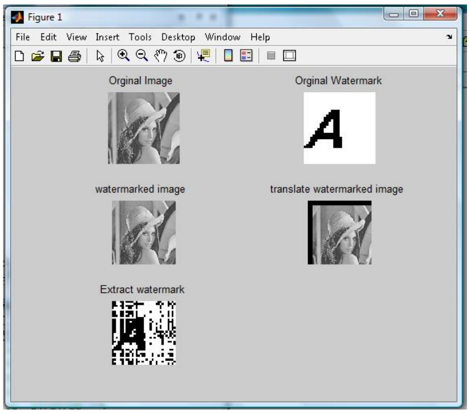
attacked by resizing from 256*256 -128*128-256*256 using FL-EXPW method



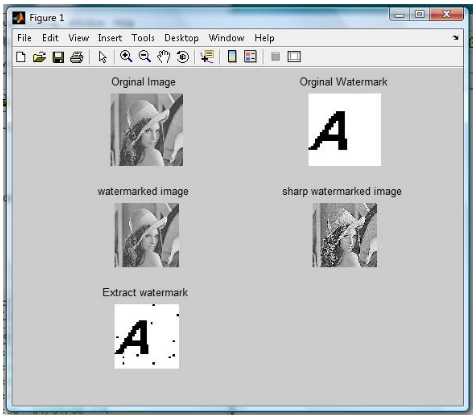
attacked by cropping using FL-EXPW method



attacked by rotating -15 degree using FL-EXPW method



attacked by translate down & right by 20 pixels using FL-EXPW method



attacked by sharpening using FL-EXPW method