Multiobjective Image Data Hiding Based on Neural Networks and Memetic Optimization

HIEU V. DANG[1], WITOLD KINSNER[1], and YINGXU WANG[2]

[1]Department of Electrical and Computer Engineering
University of Manitoba
75 Chancellor's Circle, Winnipeg, MB, R3T 5V6
CANADA
dangh@myumanitoba.ca, witold.kinsner@umanitoba.ca

[2]Department of Electrical and Computer Engineering
University of Calgary
2500 University Drive, Calgary, AB, T2N 1N4
CANADA
yingxu@ucalgary.ca

Abstract: This paper presents a hybridization of neural networks and multiobjective memetic optimization for an adaptive, robust, and perceptual data hiding method for colour images. The multiobjective optimization problem of a robust and perceptual image data hiding is introduced. In particular, trade-off factors in designing an optimal image data hiding to maximize the quality of watermarked images and the robusteness of watermark are investigated. With the fixed size of a logo watermark, there is a conflict between these two objectives, thus a multiobjective optimization problem is introduced. We propose to use a hybrid between general regression neural networks (GRNN) and multiobjective memetic algorithms (MOMA) to solve this challenging problem. Specifically, a GRNN is used for the efficient watermark embedding and extraction in the wavelet domain. Optimal watermark embedding factors and the smooth parameter of GRNN are searched by a MOMA. The experimental results show that the propsed approach achieves adaptation, robustness, and imperceptibility in image data hiding.

Key–Words: Information hiding; image data hiding; image watermarking; multiobjective optimization; memetic optimization; general regression neural networks; wavelet transforms; human visual system; quality metrics.

1 Introduction

Data hiding is the technique of embedding information (watermark) into a carrier signal (video, image, audio, text) such that the watermark can be extracted or detected later for copyright protection, content authentification, identity, fingerpringing, access control, copy control, and broadcast monitoring [1]. The important requirements for the data hiding systems are robustness, transparency, capacity, and security under different attacks and varying conditions [2, 3]. These requirements can vary under different applications. Consequently, a good data hiding technique should be adaptive to the environment. A more advanced approach should involve perception, cognition, and learning [4, 5]. In general, digital data hiding can be categorized into two classes, depending on the domain of embedding the watermark [1], (*i*) spatial domain data hiding, and (*ii*) transformed domain data hiding. Digital data hiding techniques are also classified based on the watermark data embedded into the host signal. A logo data hiding technique requires a visual watermark like a logo image, while a statistical data technique requires a statistical watermark like a pseudo random sequence. In statistical data hiding approaches (eg., [6, 7, 8]), watermarks are detected by statistical method to demonstrate that the watermark in the host signal is unchanged. In logo data hiding (eg., [9, 10, 11]), visual watermarks are extracted from the host signals for visual copyright proofs. These watermarks are not only assessed by machines but also by humans through their ability to recognize visual patterns through *human visual system* (HVS). Thus, the presentation of a visual watermark is much more persuasive than a numerical value of a statistical watermark.

Transparency and robustness are two main challenges in logo data hiding techniques since the logo consists of much information that is not easy to embed perceptually into a host signal. Moreover, the robustness in logo data hiding is so strict because it requires satisfactory recognition from human beings. With a fixed size of a logo watermark, there is a conflict between the transparency and robustness of the watermark. Increasing the transparency of watermark (or the quality of the watermarked image) decreases the robustness of the watermark and vice versa. A good logo data hiding is a robust data hiding with the acceptable quality of watermarked image. Thus, an optimal logo data hiding should be modeled as a multiobjective optimization problem.

Recently, some researchers have applied computational intelligence to design perceptual and robust data hiding systems such as *backpropagation neural networks* (BPNN) based watermarking [2, 9, 12], *support vector machine* (SVM) based watermarking [13, 14, 15], and *genetic algorithms* (GA) based watermarking [16, 17, 18], which can detect or extract the watermark without requiring the original signal for comparison. BPNNs have been recently exploited for intelligent watermarkig methods [2, 9]. The BPNNs have been used to extract the relationships between selected pixels or selected transformed coefficients and their neighbours for embedding and extracting the watermark bits. Thus, these algorithms are robust to the amplitude scaling and a number of other attacks. However, one key disadvantage of the BPNN is that it can take a large number of iterations to converge to the desired solution [19, 20]. The data hiding problems have been recently considered as single optimization problems. Shieh and coworkers [16] introduced a watermarking technique that use a GA to find the optimum frequency bands for embedding watermark bits into *discrete cosine transform* (DCT) coefficients that can improve imperceptibility or robustness of the watermark.

In this paper, an optimal logo data hiding for colour images is formulated as a multiobjective optimization problem. To solve this problem, we propose a novel logo data hiding method based on wavelets, and the hybrid of a *general regression neural network* (GRNN) and a *multiobjective memetic algorithm* (MOMA). This new method is different from previous techniques in that it utilizes a GRNN to extract relationships between wavelet coefficiets of the Y channel of the corresponding YCrCb image for embedding and extracting the watermark. Embedding factors (watermarking strenghs) and GRNN's smooth parameter are searched optimally by a MOMA to maximize the quality of the watermarked image and the robustness of the watermark. The main contributions of this work are as follows:

1. A novel logo data hiding method for colour images is proposed based on wavelets and GRNN. The optimality of the method is achieved by using a MOMA. This is the first MOMA-based approach to optimize a logo data hiding for colour images.
2. Different classes of wavelet bases are analyzed experimentally to select an appropriate wavelet for robust and perceptual data hiding techniques based on computational intelligence; and
3. A multiobjective optimization problem of logo data hiding for colour images is introduced.

The paper is organized as follows: In Sec. 2, the backgrounds of information hiding, GRNN, and MOMA are discussed. The proposed watermark embedding and extraction algorithms are introduced in Sec. 3. The optimal data hiding using MOMA is described in Sec. 4. Experimental results and discussions are given in Sec. 5.

2 Background on Methods Used

2.1 Theory of Information Hiding

2.1.1 Communication Model of Information Hiding

Information hiding can be considered as a basic communication theoretical model [21, 22, 23, 24, 25]. Cox and coworkers [22] suggested that information hiding closely resembles communications with side information at the transmitter and decoder, a configuration originally described by Shannon. Moulin *et al.* [25, 23] formulated the information hiding problem as a communication problem where the hiding capacity is considered as the maximum rate of reliable communication through the communication system. A game theory approach was proposed to seek an upper bound of the hiding capacity. In this work, we use the theory of bags, as described by [26, 27], to explain the communication model of information hiding as depicted in Fig. 1.

In this model, we denote bag $\mathbb{S}$ as the host signal, bag $\mathbb{M}$ as the hidden message (watermark), bag $\mathbb{K}$ as the secret key shared between the encoder and the decoder, bag $\mathbb{U}$ as the embedded signal, bag $\mathbb{V}$ as the received (attacked) signal, and bag $\hat{\mathbb{M}}$ as the exracted hidden message.

At the encoder, there are three inputs (three subbags) of $\boldsymbol{s}_{\mathbb{S}} \subset \mathbb{S}$, $\boldsymbol{m}_{\mathbb{M}} \subset \mathbb{M}$, and $\boldsymbol{k}_{\mathbb{K}} \subset \mathbb{K}$. The message $\boldsymbol{m}_{\mathbb{M}}$ is first scrambled with the secret key $\boldsymbol{k}_{\mathbb{K}}$ which is independent of the host signal $\boldsymbol{s}_{\mathbb{S}}$, then embedded into the host signal $\boldsymbol{s}_{\mathbb{S}}$ to produce the embedded signal $\boldsymbol{u}_{\mathbb{U}} \subset \mathbb{U}$ using an embedding function $\boldsymbol{u}_{\mathbb{U}} = \vartheta(\boldsymbol{s}_{\mathbb{S}}, \boldsymbol{m}_{\mathbb{M}}, \boldsymbol{k}_{\mathbb{K}})$.

In transit, the embedded signal $\boldsymbol{u}_{\mathbb{U}}$ is influenced by intended or unintended interference such as noise

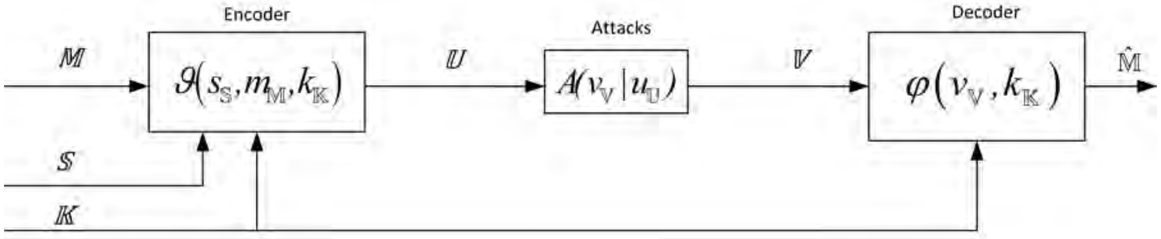
Encoder
Attacks
Decoder
$\mathbb{M}$
$\vartheta(s_{\mathbb{S}}, m_{\mathbb{M}}, k_{\mathbb{K}})$
$\mathbb{U}$
$A(v_{\mathbb{V}} | u_{\mathbb{U}})$
$\mathbb{V}$
$\varphi(v_{\mathbb{V}}, k_{\mathbb{K}})$
$\hat{\mathbb{M}}$
$\mathbb{S}$
$\mathbb{K}$

Figure 1: Communication theoretic model of information hiding.

addition, compression, filtering, amplitude scaling, and block-lost. These interferences are all considered as attacks created by attackers. An attacker takes the embedded signal $\boldsymbol{u}_{\mathbb{U}}$, and creates a modified signal $\boldsymbol{v}_{\mathbb{V}}$ by the function $\boldsymbol{v}_{\mathbb{V}} = A(\boldsymbol{v}_{\mathbb{V}}|\boldsymbol{u}_{\mathbb{U}})$. The attacker usually wants to produce the modified signal $\boldsymbol{v}_{\mathbb{V}}$ that is perceptually close to $\boldsymbol{u}_{\mathbb{U}}$, but destroys the hidden message in $\boldsymbol{u}_{\mathbb{U}}$.

At the decoder, the message $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}$ is extracted from the received (attacked) signal $\boldsymbol{v}_{\mathbb{V}}$ and the secret key $\boldsymbol{k}_{\mathbb{K}}$ by using the extracting function $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}} = \varphi(\boldsymbol{v}_{\mathbb{V}}, \boldsymbol{k}_{\mathbb{K}})$.

2.1.2 Important Technical Issues of Information Hiding

The technical issues presented here are usually considered as requirements for an information hiding technique for a specific application. They include:

a) *Transparency*: In most applications, the embedded signal $\boldsymbol{u}_{\mathbb{U}}$ is required to be received perceptually as the host signal $\boldsymbol{s}_{\mathbb{S}}$. This means that the hidden message $\boldsymbol{m}_{\mathbb{M}}$ should be invisible in the host signal $\boldsymbol{s}_{\mathbb{S}}$. The transparency is measured by comparing the two signal $\boldsymbol{u}_{\mathbb{U}}$ and $\boldsymbol{s}_{\mathbb{S}}$ using the function $t = sim(\boldsymbol{u}_{\mathbb{U}}, \boldsymbol{s}_{\mathbb{S}})$. In practice, embedding a message into a host signal always creates a distortion, d, to the signal $\boldsymbol{s}_{\mathbb{S}}$. A perceptual information hiding should minimize the distortion d regarding the human visual system to obtain the maximum transparency t.

b) *Robustness*: Robustness refers to the ability of the hidden message (watermark), which is embedded into the host signal by an information hiding technique, to survive common attacks such as signal processing operations (compression, filtering, noise addition, desynchronization, cropping, insertions) [28, 23]. The robustness of an information hiding system is measured by comparing the accuracy of the extracted hidden message $\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}$ to the hidden message $\boldsymbol{m}_{\mathbb{M}}$ by the function $a = comp(\hat{\boldsymbol{m}}_{\hat{\mathbb{M}}}, \boldsymbol{m}_{\mathbb{M}})$. There is a trade-off between the robustness and the transparency in an information hiding system.

c) *Capacity*: This refers to the number of bits of the hidden message $\boldsymbol{m}_{\mathbb{M}}$ that are able to be perceptually embedded into the signal $\boldsymbol{s}_{\mathbb{S}}$ by an information hiding technique. There is also a trade-off between the capacity and the transparency.

d) *Security*: In the worst case, when a pirate or attacker can extract the hidden message from the embedded signal $\boldsymbol{u}_{\mathbb{U}}$, the security guarantees that the pirate is not able to understand the extracted hidden message. In other words, security is the ability of the hididng algorithm to make the hidden message incomprehendible to the pirates/attackers. To have security, the hidden message $\boldsymbol{m}_{\mathbb{M}}$ is scrambled or encrypted by a scrambling or encryption technique with a secret key $\boldsymbol{k}_{\mathbb{K}}$ before being embedded into the host signal $\boldsymbol{s}_{\mathbb{S}}$.

e) *Detectability*: This refers to the ability of the hiding technique that makes the hidden message transparent to detection techniques given by the third parties. One might confuse the detectability with transparency. While the transparency refers to the transparency of the hidden message to the human perception, the detectability refers to the transparency of hidden message to the detection techniques such as statistical detection techniques. The detectability is an important requirement for steganography applications.

2.2 General Regression Neural Networks

Artificial neural networks are models inspired by the working of the human brain. They are set up with some unique attributes such as universal approximation (input-output mapping), the ability to learn from and adapt to their environment, and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data [19]. A neural network can provide an approximation to any function of the input vector, provided the network a sufficient number of nodes [29]. Because of those universal features, neural networks are studied extensively for applications in classification, pattern recognition, forecasting, process control, image compression, and others. Various classes of neural networks such as perceptron networks, multilayer perceptron networks, radial-basis function networks, self-organizing map networks, recurrent networks, and probabilistic networks have been proposed and used extensively [19]. In this section, we will provide a brief overview of the GRNN.

The GRNN, proposed by Specht [20], is a special network in the category of probabilistic neural networks (PNN). GRNN is an one-pass learning algorithm with a highly parallel structure. Different from other probabilistic neural networks, GRNNs provide estimates of continuous variables, and converges to the underlying (linear or nonlinear) regression surface. This makes GRNN a powerful tool to do predictions, approximation, and comparisons of large data sets. It also allows to have fast training and simple implementation. GRNN is sucessfully applied for image quality assessment [30], function approximation [31], and web-site analysis and categorization [32].

A diagram of the GRNN is shown in Fig. 2. In this diagram, a simple example of a one-dimensional (1D) input vector $\boldsymbol{X}[1, Q]$ is used to explain the calculation principle of the network. With the input of multidimensional vectors (i.e., matrices), it is considered as the vector of one dimensional vectorS. The network has Q neurons at the input layer, Q neurons at the pattern layer, two neurons at the summation layer, and one neuron at the output layer. The input units are the distribution units. There is no calculation at this layer. It just distributes all of the measurement variable $\boldsymbol{X}$ to all of the neurons in the pattern units layer. The pattern units first calculate the cluster center of the input vector, $\boldsymbol{X}^i$. When a new vector $\boldsymbol{X}$ is entered to the network, it is subtracted from the corresponding stored cluster center. The square differences d_i^2 are summed and fed into the activation function $f(x)$, and are given by

$$d_i^2 = (\boldsymbol{X} - \boldsymbol{X}^i)^T * (\boldsymbol{X} - \boldsymbol{X}^i) \tag{1}$$

$$f_i(\boldsymbol{X}) = \exp\left(-\frac{d_i^2}{2\sigma^2}\right) \tag{2}$$

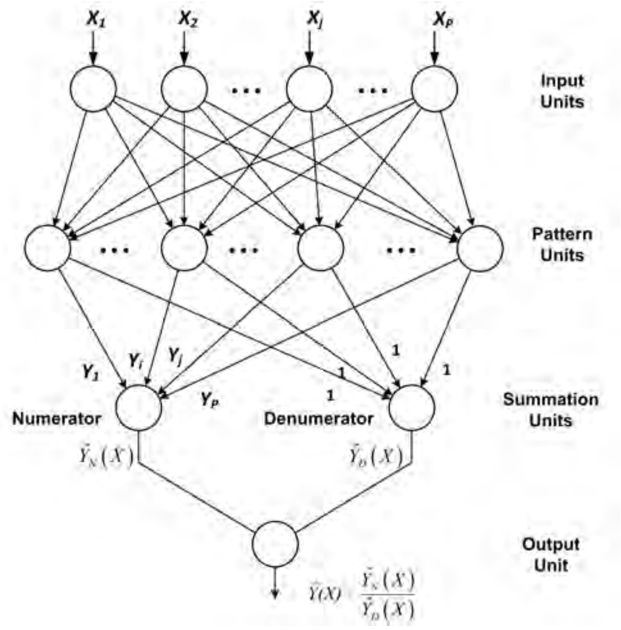

Figure 2: GRNN block diagram.

The signal of a pattern neuron i going to the numerator neuron is weighted with corresponding values of the observed values (target values), Y_i, to obtain the output value of the numerator neuron, $\hat{Y}_N(\boldsymbol{X})$. The weights of the signals going to the denumerator neuron are one, and the output value of the denumerator neuron is $\hat{Y}_D(\boldsymbol{X})$. The output value of the GRNN is the division of $\hat{Y}_N(\boldsymbol{X})$ and $\hat{Y}_D(\boldsymbol{X})$. This can be written as

$$\hat{Y}(\boldsymbol{X}) = \frac{\sum_{i=1}^{Q} Y_i f_i(\boldsymbol{X})}{\sum_{i=1}^{Q} f_i(\boldsymbol{X})} \tag{3}$$

where

$$\hat{Y}_N(\boldsymbol{X}) = \sum_{i=1}^{Q} Y_i f_i(\boldsymbol{X}) \tag{4}$$

$$\hat{Y}_D(\boldsymbol{X}) = \sum_{i=1}^{Q} f_i(\boldsymbol{X}) \tag{5}$$

In GRNN, only the standard deviation or a smooth parameter, σ, is subject to a search. To select a good value of σ, Specht recommends the use of the holdout method [20]. In our work, the optimal σ is searched by a multiobjective memetic algorithm (MOMA) for a perceptual and robust logo image watermarking.

2.3 Multiobjective Memetic Algorithms

Evolutionary algorithms (EA), like *genetic algorithms* (GA), are search-and-optimization techniques that work on a principle inspired by the natural evolution theory of Darwin [33, 34]. Inspired by models of adaptation in natural systems that combine the evolutionary adaptation of a population with individual learning within the lifetimes of its members, *memetic algorithms* (MA) have been introduced by Moscato [35] as extentions of EA that adopt the hybridization between EA and local searches to refine the individuals under consideration [33, 35]. The use of MA for multiobjective optimization (*Mutiobjective memetic algorithms* - MOMA) has attracted much attention and efford in recent years. In the literature, MOMA has been demonstrated to be much more effective and efficient than the EA and the tranditional optimization searches for some specific optimization problem domains [33, 36, 37, 38, 39]. The reports on the applications of MOMA to real engineering problems are still limited in the literature.

In multiobjective optimization problems, the solution is a family of points known as a Pareto-optimal set (i.e., Pareto solution set), where each objective component of any member in the set can only be improved by degrading at least one of its other objective components [40, 41]. The values of objectives of the Pareto solutions in the Pareto-optimal set form a Pareto front. Multiobjective optimization algorithms can be categorized into two groups: (*i*) algorithms that use the combinations of objectives to select new individuals; (*ii*) algorithms that do not combine objectives and do the selection by means of dominance based criterion [37]. In the first category, the multiple objectives are combined to create a single objective by adopting a weight values. Thus, the algorithm does not detect a Pareto front, but only one solution. This class of algorithms has the drawback that the selection of a proper set of weights must be performed to allow a natural dispersion of the solutions. In the second approach, the selection is based on dominance-based rankings of all the solutions of the population. A multiobjective memetic optimization can be developed based on the combination of objectives for a single objective optimization, or based on dominance-based criteria. In this work, we focus only on the second approach that is based on the Pareto front criteria for effective MOMA.

The performance of MOMA not only relies on the evolutionary framework, but also depends on the local search. The best tradeoff between a local search and the global search provided by evolution is the foremost issue in MOMA [33]. There are different MOMA frameworks introduced in the literature for domain-specific applications [42, 43]. Ishibuchi et al. [36] introduced a MOMA framework for combinatorial optimization problems. This work adopts a hybridization of the multiobjective genetic algorithm NSGA-II introduced by Deb and coworkers [44] and a local search to produce a MOMA for the Knapsack combinatorial optimization problem. In this work, a local search is employed to refine the offsprings with a weighted sum-based scheme. The selection criterion are based on Pareto ranking and crowding distance sorting used in NSGA-II. Motivated by the work of Ishibuchi et al. [36], the framework of the MOMA for our data hiding problem is described in the Algorithm 1.

Algorithm 1 Multiobjective Memetic Algorithm (MOMA)

```
procedure MOMA($N, p_{ls}$)
    Generate Random Population $P$ size $N$
    Objectives Evaluation
    Fast Non-Dominated Sort
    Crowding Distance Assignment
    repeat
        Generate Offspring Population $P_{offs}$
        $P_{impr}$ ←Local-Search($P_{offs}, p_{ls}$)
        $P_{inter} \leftarrow P \cup P_{offs} \cup P_{impr}$
        Fast Non-Dominated Sort
        Crowding Distance Assignment
        Update Population $P \leftarrow$ Selection($P_{inter}$)
    until Terminated Conditions
    return Non-Dominated Population $P$
end procedure
```

Algorithm 1 is a hybrid between NSGA-II and a local search. The procedures "Fast Non-Dominated Sort", "Crowding Distance Assignment" are parts of the NSGA-II described in details in [44, 34]. The procedure "Generate Offspring Population" is genetic operation procedure consisting of crossover and mutation operations. In this application, we use the real-coded crossover algorithm with probability p_x, and real-coded mutation with probability p_m [45, 44]. The offsprings are refined by the local search with probability of p_{ls}. In the local search, we use weighted-sum fitness as being recommended by Ishibuchi et al. [36]. The k objectives $(f_1, f_2, ...f_k)$ are weighted to be a single objective by

$$f(x) = \sum_{i=1}^{k} \lambda_i f_i(x) \tag{6}$$

where $(\lambda_1, \lambda_2, ..., \lambda_k)$ are random normalized weights generated according to [46]

$$\begin{cases} \lambda_1 = 1 - \sqrt[k-1]{rand()} \\ \dots \\ \lambda_j = (1 - \sum_{l=1}^{j-1} \lambda_l)(1 - \sqrt[k-1-j]{rand()}) \\ \dots \\ \lambda_k = 1 - \sum_{l=1}^{k-1} \lambda_l \end{cases} \tag{7}$$

The local search procedure is performed only on the best individuals of a given offspring generation.

Firstly, a random weight vector is generated by Eq. (7). Based on the generated random weights, the initial solution for local search is selected from offspring population using tournament selection with replacement. The same random weights are then used for the local search to produce improved population P_{impr} from selected initial indivisual. The intermediate population P_{inter} is produced by combining the current population P, the offspring population P_{offs}, and the improved population P_{impr}. The non-dominated population P is finally updated by the selection with replacement based on the Pareto ranks and crowding distances. The algorithm finishes when it meets certain terminated conditions such as predefined number of iterations. The details of applying this MOMA to our multiobjective image data hiding optimization problem are discribed in the next sections.

3 Watermark Embedding and Extraction Algorithms

3.1 Selection of Wavelets

Many wavelet-based data hiding methods have been introduced in the literature. Wavelets are widely used for data hiding because wavelet decomposition is considered to closely mimic the HVS's structure in perception [47, 48]. Extensive experimental research about the HVS has been conducted by visual psychologists over the years. They discovered that the human eye filters the image into a number of bands, each approximately one octave wide in frequency [47]. A wavelet transform is very suitable for identifying the disturbed areas where tamperings can be hidden more easily. This property allows one to exploit the HVS frequency masking effect for a perceptual data hiding. Each wavelet-based data hiding algorithm usually uses its own specific class of wavelets and decomposition level. The questions of what are the optimal wavelets and what is the sufficient level of decomposition for image data hiding are still open-ended.

In this work, we investigate 36 wavelet functions in 5 wavelet families for image data hiding in connection to computational intelligence-based data hiding algorithm. They are Haar (known as Db1), Daubechies (Db2, Db3, Db4,..., Db10), Symlets (Sym2, Sym3, ... , Sym8), Coiflets (Coif1, Coif2, ..., Coif5), and Biorthogonal (Bior1.3, Bior1.5, Bior2.2, ..., Bior6.8) wavelets. The test algorithm WAT-GRNN for embedding and extracting the watermark in wavelet domain will be discribed in details in Secs. 3.2 and 3.3. In the first stage, we do simulations and comparisons for wavelet functions in each wavelet family. The experimental benchmark consists of a quality (transparency) test and common attacks such as noise addition, JPEG compression, filtering, cropping, amplitude scaling. We then select the wavelets that produced better results, and compare them together. Table 1 show the *peak signal to noise ratio* (PSNR) of the watermarked images of the Lena image using these better wavelets in the case of using embedding factor of $\eta = 18$. An example of the robustness of watermark against Gaussian noise attack is depicted in Fig. 3. All wavelets used in these experiments are decomposed in 4 level. Based on these simulations, Sym2 wavelet offers us a better robustness and an acceptable quality for the watermarked image.

Table 1: PSNR of Watermarked Image using Different Wavelets for Lena image

Wavelets	db2	db4	sym2	bior1.3	coif2
PSNR (dB)	42.25	42.66	42.46	41.97	42.81

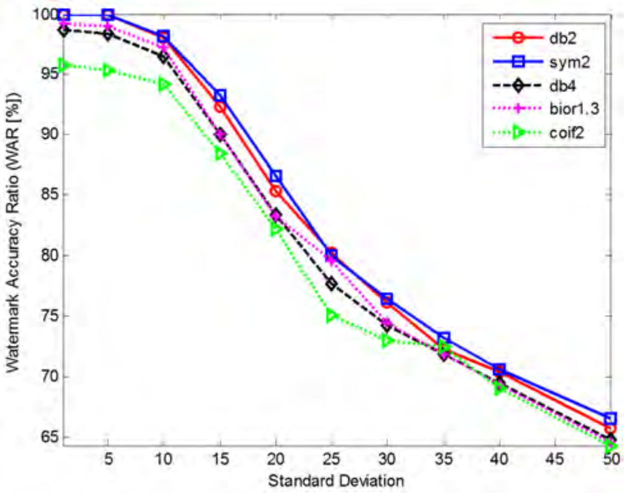

Figure 3: Robustness of wavelets against Gaussian noise addition attacks for Lena colour test image

The wavelet Sym2 is then selected to implement the algorithm with three different levels of decomposition. It can be seen from Fig. 4 that four-levels of decomposition provides a better robustness against Gaussian-noise attacks when compared to two-levels and three-levels of decomposition. The same behavior to JPEG compression, filterings, amplitude scaling, cropping attacks are also observed. From the above results, we choose Sym2 wavelet and four-levels of decomposition as the appropriate wavelet decomposition tool for our logo data hiding approach in this paper.

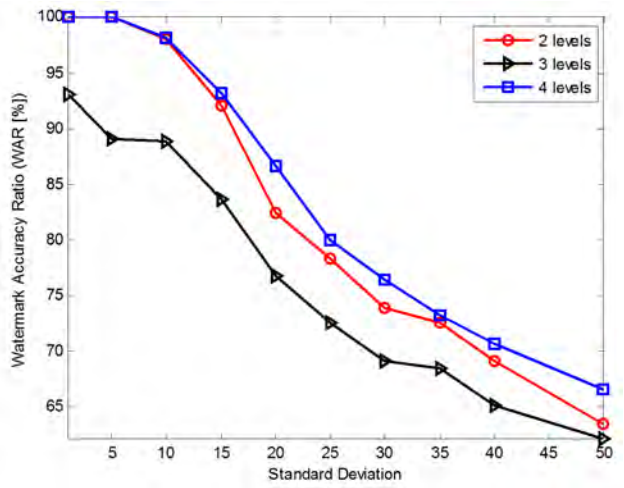

Figure 4: Robustness of the Sym2 wavelet against Gaussian noise addition attacks in different decomposition levels for Lena colour test image.

3.2 Watermark Embedding Algorithm

The proposed watermark embedding scheme is depicted in the Fig. 5. In this work, we use an RGB colour image as the host image. The watermark image is a binary logo image. The RGB image is first converted to YCrCb colour image. The luminance component Y is decomposed by wavelet transform. In this paper, we only select the luminance component Y of YCbCr colour image for embedding the watermark because of the following reasons: (i) colour channels Cr and Cb have so much redundant information for HVS so that compression techniques for colour images do most compression work in these colour channels (hence, embedding watermark in CrCb will create more redundancy and make watermark susceptible to compression attacks); (ii) luminance Y is more sensitive to HVS that any tampering is easily detected (this makes data hiding in Y channel more robust than hiding in color channels CrCb). The wavelet coefficients in each band are grouped into 3-by-3 non-overlapping blocks. Based on the random number sequence generated from the key (i, p), the algorithm selects blocks of wavelet coefficients for embedding the watermark. These wavelet coefficients in the selected blocks are used to train the GRNN. The watermark bits are embedded into selected coefficients by training the GRNN. Finally, the inverse wavelet transform IDWT is applied to reconstruct the watermarked image.

The Y component is decomposed by Symlet-2 $(sym2)$ DWT in four-levels, as shown in Fig. 6. The watermark bits are embedded only into the following subbands: HL^4, LH^4, HH^4, HL^3, LH^3, HH^3, HL^2, LH^2, HH^2, HL^1, LH^1. In our scheme, scaling coefficients in LL^4 and coefficients in HH^1 are not used for embedding the watermark since embedding in LL^4 will degrade the watermarked image, while embedding the watermark in subband HH^1 will make the watermark more susceptible. These selected subbands are divided into non-overlapping 3-by-3 blocks and then scanned to arrange into a sequence of blocks with the subband order $HL^4LH^4HH^4HL^3LH^3HL^2LH^2HH^2HL^1LH^1$. The blocks for embedding watermarks are then selected randomly by the sequence of random non-repeated integer numbers generated by the Fibonacci p-code algorithm [49] using the key (i, p). The relationship between wavelet coefficients and its neighborhoods in selected 3-by-3 blocks are extracted by a given GRNN for watermark embedding and extracting processes. The Fibonacci p-code sequence is defined by [49]

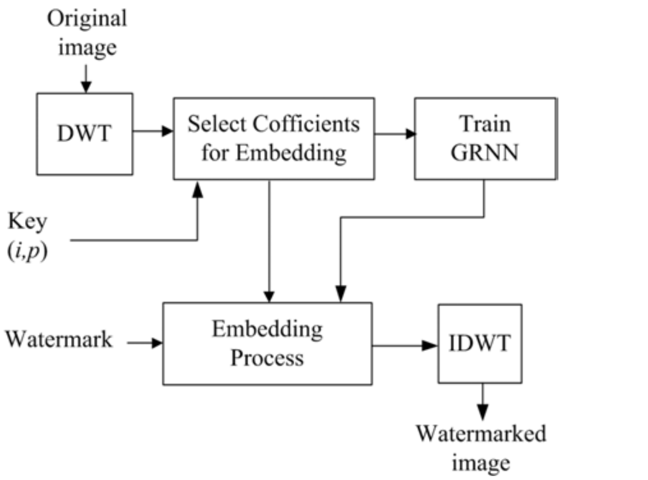

Figure 5: Block diagram of the proposed watermark embedding scheme.

$$F_p(n) = \begin{cases} 0 & \text{if } n = 0, \\ 1 & \text{if } n = 1, \\ F(n-1) + F(n-p-1) & \text{if } n > 1, p \in Z^+ \end{cases} \tag{8}$$

Then for K sequence (k=1,2,...,K), the sequence of random integer numbers $T_k = T_1, T_2, ..., T_K$ is generated by

$$T_k = k(F_p(n) + i) \bmod F_p(n+1) \tag{9}$$

where $k = 1, 2, 3, ..., K$; $i \in [-3, 3]$ and i is an integer such that $F_p(n)+i < F_p(n+1)$. The security key or the key to generate K non repeated random integer numbers are parameters (i, p).

We now have selected blocks for embedding watermark bits. With each block B_i having the center coefficient $I(i, j)$, the iput vector X_i and target T_i are

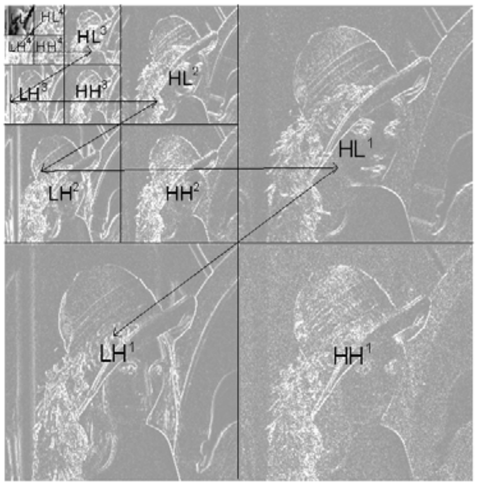

Figure 6: Intensity-adjusted display of 4-level wavelet decomposition of Lena colour image (wavelet subbands are rescaled to a gray-intensity range for display), and the scanning order of subbands for watermarking.

set up as in Eq. (10) to train the GRNN with 8 input neurons, 8 pattern neurons, 2 summation neurons, and 1 output neuron. Where $i = 1, 2, ..., K$; K is the number of watermark bits.

$$\begin{cases} X_i = \big[I(i-1,j-1), I(i-1,j), I(i-1,j+1), \\ \qquad I(i,j-1), I(i,j+1), I(i+1,j-1), \\ \qquad I(i+1,j), I(i+1,j+1)\big] \\ T_i \ = [I(i,j)] \end{cases} \tag{10}$$

With each pair (X_i, T_i), the GRNN produces the ouput $\hat{I}(i,j)$. The watermark bits are embedded into the selected block-center coefficients according to

$$I_w(i,j) = \hat{I}(i,j) + \eta(i)(2W(i) - 1) \tag{11}$$

where $\eta(i)$ is the embedding factor for each embedding watermark bits to selected block-center coefficient $I(i,j)$ of selected block B_i. They can be altered to obtain the imperceptibility and robustness. If η is small, we get a higher-quality watermarked image, but lower level of robustness, and vice versa. This is a trade-off between the quality of the watermarked image with the robustness of watermark. $W(i)$ is the i^{th} watermark bit in the sequential watermark bits. $I_w(i,j)$, the watermarked coefficient, is obtained by replacing the central coefficient $I(i,j)$ by the combination of the output of the GRNN $\hat{I}(i,j)$ and the watermark bit $W(i)$. After embedding, an inverse DWT is performed to get the watermarked luminance Y. By combining the watermarked Y with Cr, Cb and converting to RGB, the colour watermarked image is reconstructed. This embedding algorithm is denoted as WAT-EMB procedure.

3.3 Watermark Extraction Algorithm

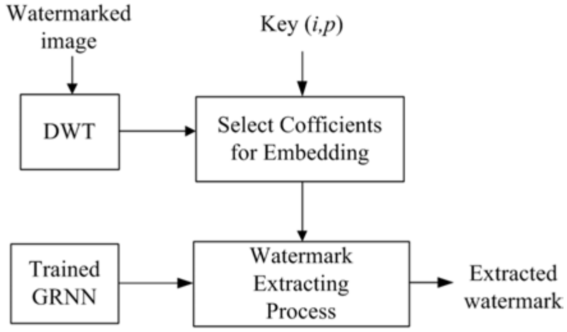
Watermarked image
Key (i,p)
DWT
Select Cofficients for Embedding
Trained GRNN
Watermark Extracting Process
Extracted watermark

Figure 7: Block diagram of the proposed watermark extraction scheme.

The watermark extraction scheme is illustrated in Fig. 7. The extraction process is the inverse of the embedding process. The colour watermarked image is first converted to $YCrCb$ colour domain. The luminance Y is then decomposed by 4-level Symlet-2 DWT. The wavelet coefficients are grouped into 3-by-3 blocks and arranged into the ordering sequence as described in Sec. IIIB. From the key (i,p) received, the sequence of random integer numbers are generated based on the Fibonacci p-code algorithms to detect the watermarked blocks. Denote I_w is the wavelet decomposition of the component Y of the watermarked image. From the detected blocks, we setup the input vector X'_i as in Eq. (10). The trained GRNN (obtained in the embedding process) is used to extract the watermark bits. For each input vector X'_i, the trained GRNN produces the output $\tilde{I}(i,j)$. The watermark bit extraction is performed by

$$\tilde{W}(i) = \begin{cases} 1 & \text{if } I_w(i,j) \geq \tilde{I}(i,j) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where i=1,2,...,K, K is the block number, and also is the number of watermark bits. $\tilde{W}$ is the extracted watermark. The extraction algorithm is denoted as WAT-EXTR procedure.

If the data hiding algorithms described in Secs. 3.2 and 3.3 use a fixed value η and a predefined fixed value of smooth parameter of GRNN σ (e.g., $\eta = 18$, $\sigma = 0.5$), we label it as the WAT-GRNN algorithm.

4 Optimal Image Data Hiding Using MOMA

In logo data hiding, with the fixed logo watermark, there always exist two conflicting objectives. These are robustness of the watermark and quality of the watermarked image (imperceptibility or transparency of watermark). In this work, we apply MOMA to search for the optimal parameters. They are the smooth parameter of the GRNN σ, and K embedding factors $\eta(i)$, $i = (1, 2, ..., K)$. The pseudocode of the proposed algorithm is described in Algorithm 2.

Algorithm 2 WAT-MOMA

```
procedure WAT_MOMA($I, W, N, i, p, p_{ls}$)
    Generate Random Integer Numbers $RN$ from key $(i, p)$
    Generate Random Population $P$ size $N$
    $P \leftarrow$ OBJ-EVAL($P, W, I, RN$)    ▷ Evaluate Objectives
    Fast Non-Dominated Sort
    Crowding Distance Assignment
    $itrs \leftarrow 0$
    repeat
        $itrs \leftarrow itrs + 1$
        Generate Offspring Population $P_{offs}$
        $P_{offs} \leftarrow$ OBJ-EVAL($P_{offs}, W, I, RN$)
        $P_{impr} \leftarrow$ LOCAL-SEARCH($P_{offs}, p_{ls}, I, W, RN$)
        $P_{inter} \leftarrow P \cup P_{offs} \cup P_{impr}$
        Fast Non-Dominated Sort
        Crowding Distance Assignment
        Update Population: $P \leftarrow$ Selection($P_{inter}$)
    until $itrs \geq MaxItrs$
    $S_{best} \leftarrow$ Sol-Select($P$)
    $I_W \leftarrow$ WAT-EMB($BSOL, I, W, RN$)
    return $I_W$
end procedure
```

The inputs consist of the N number of chromosomes in population P, the colour image I, the watermark W, key (i, p), and the probability of the local search p_{ls}. From the key (i, p), the algorithm generates a sequence of random numbers RN based on the Fibonacci p_code algorithm from Eqs. (8) and (9). Each chromosome consists of $(1 + K)$ genes. The first genes represents for the smooth paramenter σ of the GRNN used for embedding and extracting the watermark. The next K genes represents K embedding factors $\eta(i)$ with $i = 1, 2, ..., K$, where K is the number of watermark bits embedded into the image. The procedure OBJ-EVAL is used to evaluate objectives for each chromosome in the given population. In this work, we search for optimal data hiding parameters to maximize the quality of watermarked image, and the averaged robustness of watermark in the case of noise addition attack, JPEG compression attack, amplitude scaling attacks, and filtering attacks. The Tabu local search [50] uses the random weighted fitness as described in Sec. 2.3, and the random weights obtained from Eq. (7).

The best solution or best chromosome (S_{best}) will be selected from the non dominated population P. Finally, we obtained the watermarked image I_W by implementing the watemark embedding algorithm presented in Sec. 3.2 (WAT-EMB) with smooth parameter $\sigma = S_{best}(1)$, embedding factors $\eta(i) = S_{best}(i + 1)$, $i = 1, 2, ..., K$. At the decoder side, the watermark is extracted by the watermark extraction process presented in Sec. 3.3 (WAT-EXTR). The initialization and objective evaluation algorithms are discussed as follows.

1) *Initialization*: Each chromsome represents $1 + K$ real nonegative parameters to be searched. The first parameter is smooth parameter of the GRNN, which is set in the range from 0.1 to 5. The K remaining parameters represents for the K watermarking factors $\eta(i), i = 1, 2, ..., K$. The watermarking factors are searched in a wide range from 1 to 50.

2) *Objective Function Evaluation*: In literature, the objective function is also called the fitness function. The objective function uses the *peak signal to noise ratio* (PSNR) as the quality objective, and the averaged *watermark accuracy ratio* (WAR) in the cases of four different attacks as robustness objectives. The PSNR is defined by

$$\text{PSNR} = 10 \log_{10} \left(\frac{I_{peak}^2}{\text{MSE}} \right) \tag{13}$$

where I_{peak} is the maximum intensity value of the three color channels R, G, B, and the *mean squared error* (MSE) computed for all three color channels R, G, and B is given by

$$\text{MSE} = \frac{1}{KMN} \sum_{k=1}^{3} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j,k) - I_W(i,j,k))^2 \tag{14}$$

The watermark accuracy ratio is defined by

$$\text{WAR} = \frac{\sum_{i=1}^{M_w} \sum_{j=1}^{N_w} W(i,j) \bar{\oplus} \tilde{W}(i,j)}{M_w * N_w} \tag{15}$$

where W and $\tilde{W}$ are the original and extracted watermarks, and (M_w, N_w) is the size of the watermarks. The logic operator $\bar{\oplus}$ does comparison between W and $\tilde{W}$. $W(i,j) \bar{\oplus} \tilde{W}(i,j) = 1$ if $W(i,j)$ and $\tilde{W}(i,j)$ have the exactly same value of 0 or 1. If WAR $\geq 70\%$, the extracted watermark can be considered as the original watermark. It is close to be perfect if WAR $\geq 85\%$.

Let $K = M_w * N_w$ be the number of watermark bits embedded into the image. We denote $\bar{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_{K+1}]$ as the data hiding parameters to be searched, where $\alpha_1 = \sigma$ (the smooth parameter of the GRNN), $\alpha_{2:K+1} = \eta(1 : K)$ (the embedding factors). The objectives function is then set up as follows

$$\bar{f}(\bar{\alpha}) = [f_1(\bar{\alpha}), f_2(\bar{\alpha})] \tag{16}$$

where

$$f_1(\bar{\alpha}) = \text{PSNR}(\bar{\alpha}) = \text{PSNR}(\alpha_1, \alpha_2, ..., \alpha_{K+1})$$

and

$$f_2(\bar{\alpha}) = \frac{W_G(\bar{\alpha}) + W_J(\bar{\alpha}) + W_A(\bar{\alpha}) + W_M(\bar{\alpha})}{4}$$

where W_G is the WAR in the case that the watermarked image is tampered by the Gaussian noise addition attack; W_J is the WAR under JPEG compression attack; W_A is the WAR under the amplitude scaling attack; and W_M is the WAR under the median filtering attack. Our optimal watermarking problem is to search optimal parameters $\bar{\alpha}$ that can be formed by

$$\max_{\bar{\alpha}} \bar{f}(\bar{\alpha}) = \max_{\bar{\alpha}} [f_1(\bar{\alpha}), f_2(\bar{\alpha})] \qquad (17)$$

The pseudocode of our objective function evaluation is described in Algorithm 3.

Algorithm 3 OBJ-EVAL

```
procedure OBJ-EVAL($P, I, W, R_N$)
    $N \leftarrow size(P, 1)$    ▷ Number of chromosome in population P
    for $i \leftarrow 1, N$ do
        $\sigma \leftarrow P(i, 1)$    ▷ Smooth parameter of GRNN
        $[I_W, grnn\_weight] \leftarrow$ WAT-EMB($P(i,:), I, W, R_N$)
        $f_1 \leftarrow$ PSNR($I_W, I$)
        $I_{WG} \leftarrow I_W + GaussNoise$    ▷ AWGN attack
        $\tilde{W} \leftarrow$ WAT-EXTR($I_{WG}, grnn\_weight, \sigma, R_N$)
        $W_G \leftarrow$ WAR($\tilde{W}, W$)
        $I_{WJ} \leftarrow$ JPEG($I_W$)    ▷ JPEG compression attack
        $\tilde{W} \leftarrow$ WAT-EXTR($I_{WJ}, grnn\_weight, \sigma, R_N$)
        $W_J \leftarrow$ WAR($\tilde{W}, W$)
        $I_{WA} \leftarrow$ AmplitudeScaling($I_W$)    ▷ Scaling attack
        $\tilde{W} \leftarrow$ WAT-EXTR($I_{WA}, grnn\_weight, \sigma, R_N$)
        $W_A \leftarrow$ WAR($\tilde{W}, W$)
        $I_{WM} \leftarrow$ MedianFilter($I_W$)    ▷ Median filtering attack
        $\tilde{W} \leftarrow$ WAT-EXTR($I_{WM}, grnn\_weight, \sigma, R_N$)
        $W_M \leftarrow$ WAR($\tilde{W}, W$)
        $f_2 \leftarrow (W_G + W_J + W_A + W_M)/4$
        $f(i,:) \leftarrow [f_1, f_2]$
    end for
    return $f$
end procedure
```

3) *Local Search*: In this work we employ the principle of Tabu local search [50] with random normalized weights generated by [46]. In this local search, the random normalized weights rnd_weight are generated by Lamda() from Eq. (7). The best initial solution for the local search is selected by doing a tournament selection between chromosomes in the population P_{offs}. The procedure finally returns the N_{LS} better solutions, P_{impr}.

4) *Crossover, Mutation and Selection with Replacement Operations*: Genetic operators including crossover and mutation are used to generate offspring population in each evolutionary loop. In this work, the real-coded crossover and mutation introduced in [45, 44] are adopted with crossover probability $p_x = 0.8$ and mutation probability $p_m = 0.05$. The nondominated chromosomes are selected in each evolutionary loop by using the selection with replacement based on the Pareto ranks and crowding distances as described in [44, 34].

5 Experimental Results and Discussion

In this section, experimental results are demonstrated and discussed to show the watermark robustness and transparency of the proposed algorithm. In the embedding process, the memetic algorithm is used to search for optimal embedding factors and the optimal smooth parameter of the GRNN. In the watermark extraction process, the original image is not required, but the secret key (i, p), the smooth and weight parameters of the trained GRNN from the embedding process are needed. The watermark extraction process is the same as the watermark extraction algorithm described in the Sec. 3.3 (WAT-EXTR). The experimental results obtained from the proposed algorithm using multiobjective memetic algorithm (WAT-MOMA) are compared with results of the WAT-GRNN algorithm, Kutter's method [51], and Yu's method [9]. WAT-GRNN is the watermarking algorithm used WAT-EMB in Sec. 3.2 and WAT-EXTR in Sec. 3.3 with the fixed embedding factor (embedding strength) $\eta = 18$, and the smooth parameter of the GRNN $\sigma = 0.5$. In the Yu's and Kutter's methods, we setup the watermark strength $\alpha = 0.2$ to have a good robustness to be compared to the proposed algorithm WAT-MOMA.

To evaluate the performance of our data hiding algorithms, the ”Winipeg Jet” logo is embedded into various colour images. The binary watermark of size 64-by-64 is embedded into highly-textual colour images ”Lena”, ”Baboon”, ”Airplane-F16”, and ”House” each with size of (512-by-512)-by-3.

5.1 Results of Multiobjective Memetic Optimization Algorithm

In the WAT-MOMA algorithm, which uses the multiobjective memetic optimization to search for optimal embedding factors and the smooth parameter of GRNN, the number of initial chromosomes N setup to 100, the local search is applied to refine the offspring population with the probability of 0.5 and the number of iterations is 50. These local search parameters are selected based on the analystical results shown in [36] for memetic algorithm using weighted sum-based local search.

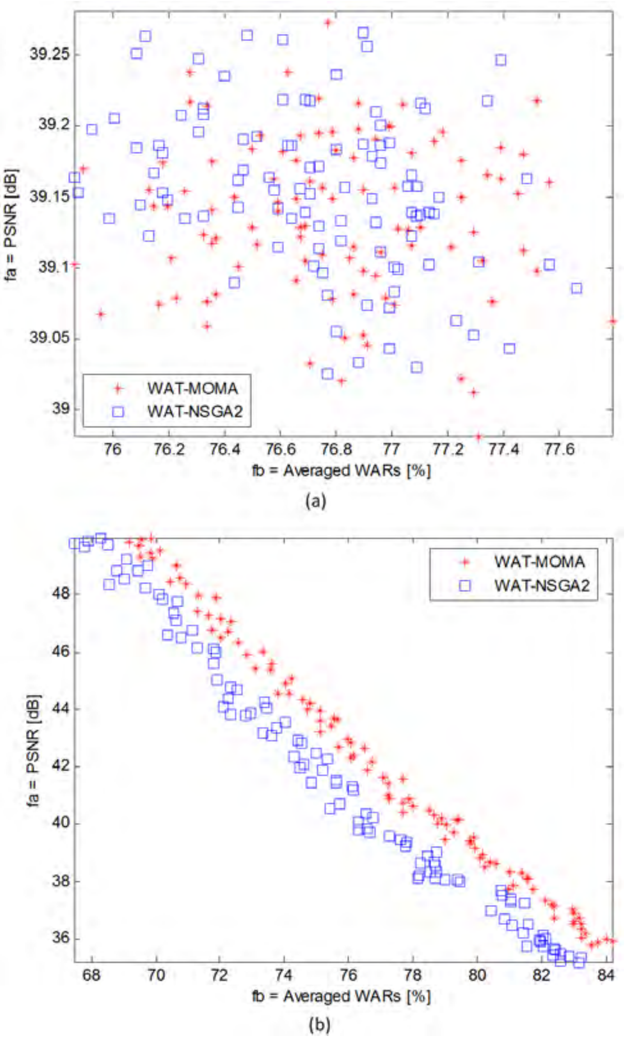

Figure 8: Numerical results of the watermarking based on memetic and NSGA2 strategies for Lena color image: (a) WAT-MOMA's initial population versus WAT-NSGA2's, (b) WAT-MOMA's population versus WAT-NSGA2's population after 100 iterations.

The numerical results in Fig. 8. shows that the algorithm based on memetic optimization is more effective than the algorithm based on multiobjective genetic algorithm NSGA-II [44]. Since there is conflict between the quality of watermarked image and the robustness of watermark in image data hiding, the optimally selected solution (chromosome) is a balance between the PSNR objective and the averaged WARs objectives. The solution includes the smooth parameter of GRNN and 64x64=4096 embedding factors. Example of the optimal embedding factors for Lena colour image after 100 iterations are illustrated in Fig. 9 corresponding the smooth parameter of GRNN $\sigma = 2.47657$.

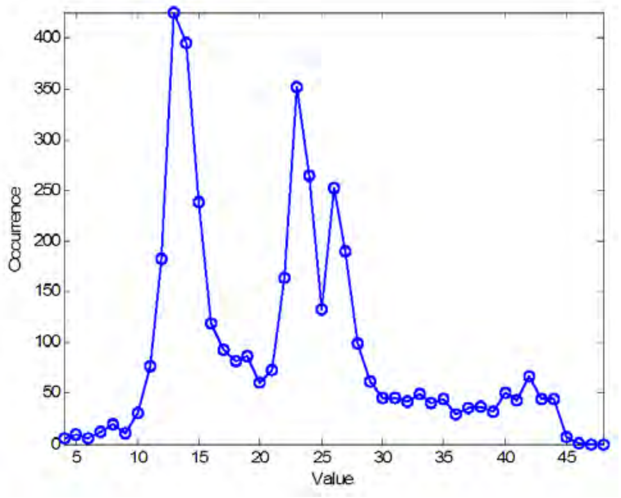

Figure 9: Watermarking factors for Lena colour image obtained after WAT-MOMA run 100 iterations.

5.2 Quality Evaluation

To measure the transparency or the similarity of the watermarked image to the original image, image data hiding systems mostly employ the PSNR. In Fig. 10, the differences between the original images and the watermarked images are difficult to observe by human eyes. The PSNRs obtained by WAT-MOMA for all these four colour test images are compared with PSNRs obtained by WAT-GRNN, Yu's method, and Kutter's method. The comparison results are described in Table 2.

Table 2: PSNR comparison of atermarked images

Images	PSNR [dB]			
	Kutter's	Yu's	WAT-GRNN	WAT-MOMA
Lena	41.8433	41.6670	42.4590	42.8180
Baboon	41.3612	41.2206	42.5781	42.4320
Airplane	38.6961	38.5295	42.3353	42.8027
House	39.4374	39.2806	42.3143	42.8596

5.3 Robustness Evaluation

The robustness of the watermark is evaluated by the similarity between the extracted watermark and the original watermark throuth WAR computed by Eq. (15). The watermarks extracted from the watermarked images in Fig. 10 are shown in Fig. 11. The calculated WARs indicate that our method perfectly extracts watermarks from watermarked images in the

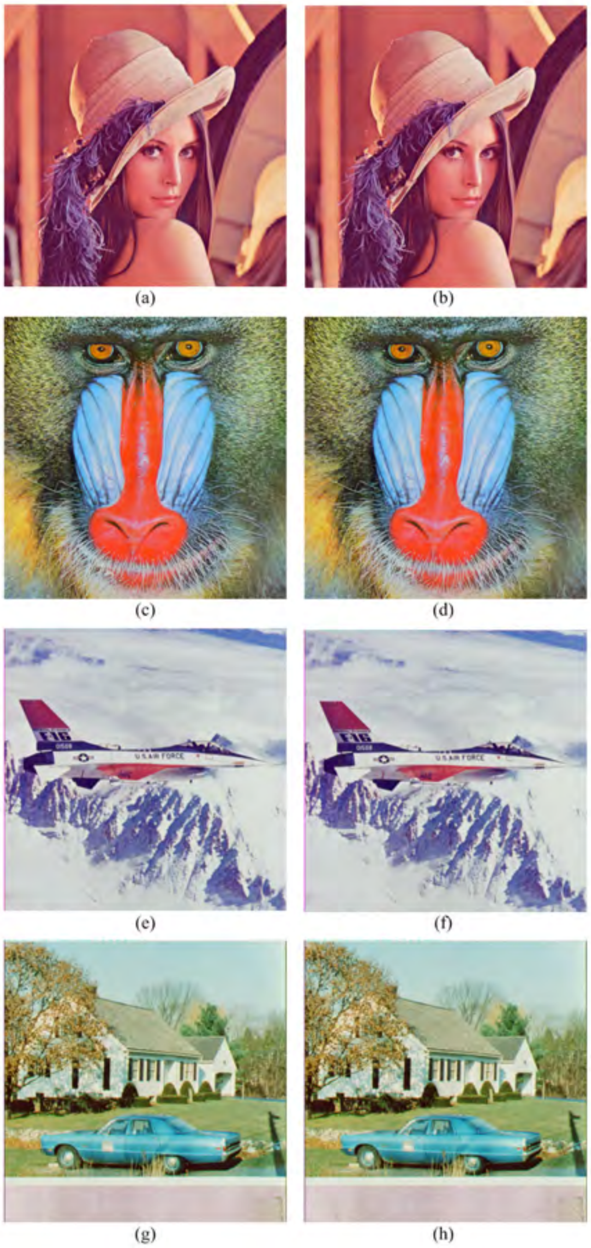

Figure 10: The original test images and watermarked test images: (a) original Lena image, (b) watermarked lena image with the obtained PSNR=42.82 dB, (c) original Baboon image, (d) watermarked Baboon image with the obtained PSNR=42.43 dB, (e) original Airplane F16 image, (f) watermarked Airplane F16 image with the obtained PSNR=42.80 dB, (g) original House image, (h) watermarked House image withe the obtained PSNR=42.86 dB.

case of without any attacks.

We test the proposed algorithm with different classes of attacks such as (*i*) compression attacks (JPEG compression), (*ii*) noise addition attacks

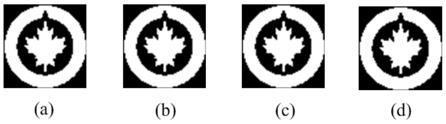

Figure 11: Watermarks extracted from watermarked images in Fig. 10: (a) extracted from Fig. 10(b) with WAR=100 %, (b) extracted from Fig. 10(d) with WAR=100 %, (c) extracted from Fig. 10(f) with WAR=100 %, (d) extracted from Fig. 10(h) with WAR=100 %.

(AWGN, salt & pepper, and fractional noises), (*iii*) filtering attacks (median filtering), (*iv*) amplitude scaling attacks.

1. *Robustness Against JPEG Compression*: JPEG is common image compression standard for multimedia application. Hence, image data hiding systems should be robust to this attack. Figure 12 shows an example of JPEG compression attack with the quality factor of 40 to the watermarked images of Lena and Baboom, and the proportional extracted watermarks. The robustness comparison with WAT-GRNN, Yu's and Kutter's methods for the watermarked image of Lena in Fig. 10 is displayed in Fig. 13.

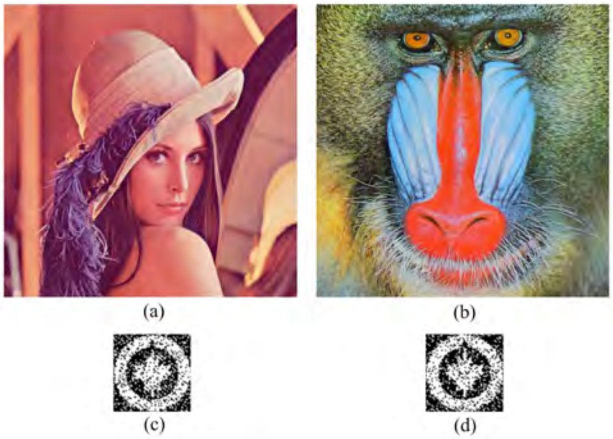

Figure 12: An example of JPEG compression attack and watermark extraction with JPEG quality factor of 40: (a) compression of watermarked image of Lena at Fig. 10(b) with SNR=26.14 dB, (b) compression of watermarked image of Baboom at Fig. 10(d) with SNR=18.98 dB, (c) the extracted watermark from (a) with WAR=82.47 %, (d) the extracted watermark from (b) with WAR=83.42 %.

2. *Robustness Against Amplitude Scaling*: The colour values of the watermarked image are divided by a *scaling factor* (SF). The attack is called negative amplitude scaling attack if SF is greater than one,

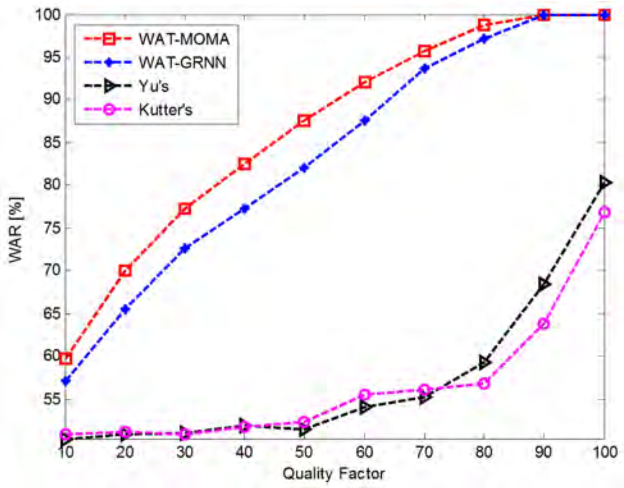

Figure 13: The experimental results under the JPEG compression attack for watermarked image of Lena.

and vice versa is the positive amplitude scaling attack. An example of the positive amplitude scaling attack with SF= 0.3 for watermarked images of Lena and Baboom in Fig. 10 are depicted in Fig. 14. The robustness of watermark compared with results from WAT-GNRR, Yu's and Kutter's methods is illustrated in Fig. 15.

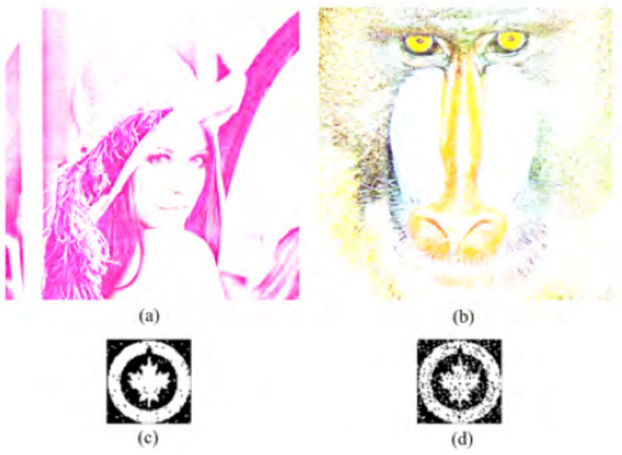

Figure 14: An example of amplitude scaling attack and watermark extraction with SF=0.3: (a) scaling the watermarked image of Lena at Fig. 10(b) with SNR=-7.36 dB, (b) scaling the watermarked image of Baboom at Fig. 10(d) with SNR=-7.36 dB, (c) the extracted watermark from (a) with WAR=98.09 %, (d) the extracted watermark from (b) with WAR=90.09 %.

It can be seen that the WAT-MOMA algorithm is very robust to amplitude scaling attacks. Even if with the positive attack of SF=0.3 that decreases the SNR of the attacked watermarked image to -7.36 dB, we are still able to recover the watermark excellently.

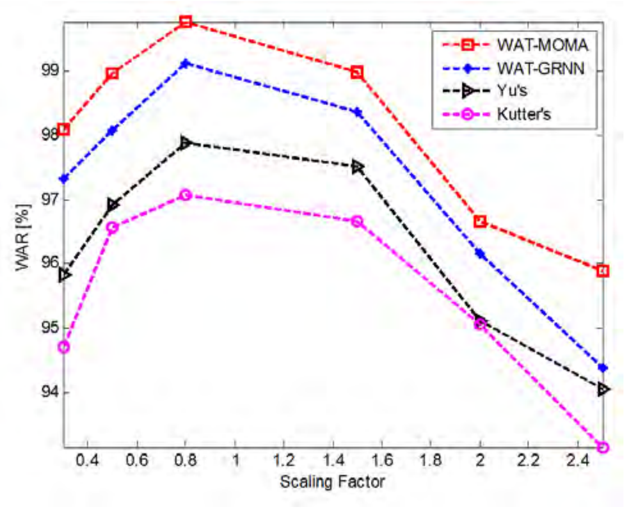

Figure 15: The experimental results under the amplitude scaling attack for watermarked image of Lena.

3. *Robustness Against Additive White Gaussian Noise*: Since the natural features of electronic devices and communications channels, AWGN is perhaps the most common noise in communications systems. Thus, a good data hiding scheme should be robust to AWGN. The robustness fo our scheme against AWGN is shown in Fig. 16 and Fig. 17.

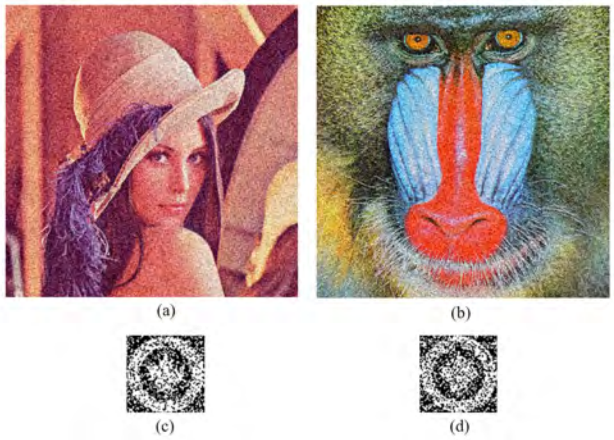

Figure 16: An example of AWGN noise attack and watermark extraction with variance of AWGN= 40^2: (a) attacked watermarked image of Lena at Fig. 10(b) with SNR= 10.9 dB, (b) attacked watermarked image of Baboom at Fig. 10(d) with SNR=10.74 dB, (c) the extracted watermark from (a) with WAR=75.34 %, (d) the extracted watermark from (b) with WAR=71.73 %.

The AWGN is added to the watermarked images with different standard deviation σ_n (corresponding

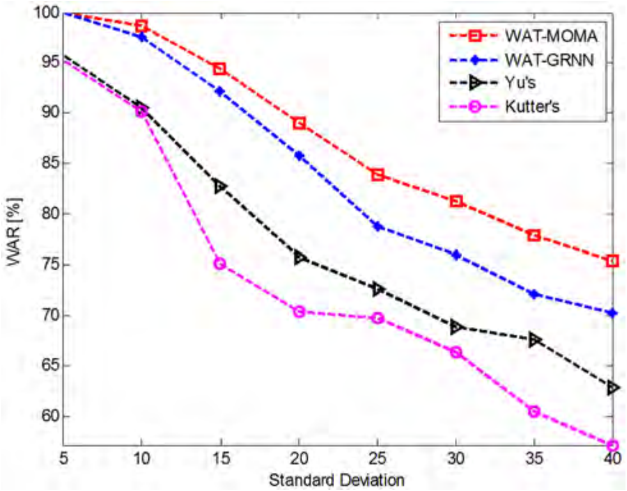

Figure 17: The experimental results under the AGWN noise attack for watermarked image of Lena.

SNRs). The Gaussian noise is added to the colour image of watermarked image, I_W, by

$$I_W^N = I_W + \sigma_n N \quad (18)$$

where N is the normally distributed random noise, and I_W^N is the watermarked image corrupted by the Gaussian noise. The proposed method works really well, even with a variance of AWGN=40^2 (with the equivalent SNR around 10 dB). This level is a challenge to every data hiding and denoising techniques [52, 53].

4. *Robustness Against Median Filtering*: Median filtering is always a serious challenge to watermarks. This is because a median filter does average pixel values in the window size that eliminates high dynamic values in the image in the spatial domain. Hence, median filtering can affect the watermark severely. An example of doing median filtering for watemarked images of Lena and Baboon with the filter window size of 5 is displayed in the Fig. 18. The robustness comparison of the proposed algorithm with other methods for the watermarked image of Lena is depicted in Fig. 19.

6 Conclusions

In this paper, a logo data hiding for colour images is formulated as a multiobjective optimization problem of finding the optimal parameters to maximize the quality of watermarked image and the robustness of the watermark under different attacks. A novel intelligent and robust logo data hiding method based on the general regression neural networks and multiobjective memetic algorithms is proposed to solve this challenging problem. Specifically, the embedding factors and the smooth parameter of the GRNN are searched optimally by the multiobjective memetic optimization to maximize the PSNR and the averaged WARs objectives. The proposed algorithm obtains better results in transparency and robustnesses against classes of additive noise, and signal processing attacks than previous approaches.

We discuss the application of neural networks for watermarking systems. We evaluated neural networks, and selected GRNN for its good fit to our problem. The GRNN is much superior over the BPNN when solving this problem as it has very fast time convergence and high prediction accuracy.

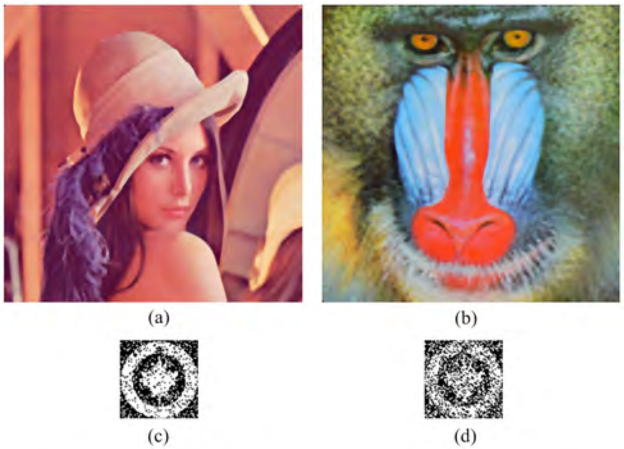

Figure 18: Median filtering attack to the watermarked images with filter window = 5: (a) filtered watermarked Lena image with SNR=26.87dB; (b) filtered watermarked Baboon image with SNR=17.43 dB; (c) watermark extracted from (a) with WAR=84.45 %; (d) watermark extracted from (b) with WAR=70.73 %.

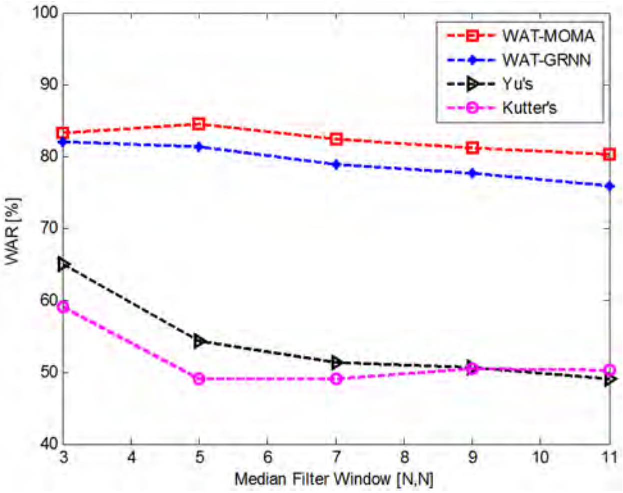

Figure 19: Experimental results under median filtering attacks for the watermarked images of Lena.

However, the proposed algorithm has its own disadvantages and needs further improvements. For example, since it needs a sufficient time for the evolutionary and local refining searches to find the best local and global solutions, it is not fast enought for real-time applications at this stage.

References:

[1] Min Wu and Bede Liu, "Data hiding in image and video: Part I - Fundamental issues and solutions," *IEEE Trans. Image Processing*, vol. 12, no. 06, pp. 685-695, June 2003.

[2] Jeng-Shyang Pan, Hsiang-Cheh Huang, and Lakhmi C.Jain, *Intelligent Watermarking Techniques*, New Jersey, MA: World Scienstific, 2004, 852 pp.

[3] Benoit Macq, Jana Dittmann, and Edward J. Delp, "Benchmarking of image watermarking algorithms for digital rights management," *Proc. IEEE*, vol. 92, no. 6, June 2004.

[4] Witold Kinsner, *Towards cognitive security systems*, in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, ICCI*CC 2012, (Kyoto, Japan; August 22-24, 2012), 2012, (Keynote Speech).

[5] Yingxu Wang, James A. Anderson, George Baciu, Gerhard Budin, D. Frank Hsu, Mitsuru Ishizuka, Witold Kinsner, Fumio Mizoguchi, Kenji Sugawara, Shusaku Tsumoto, Du Zhang, "Perspectives on eBrain and cognitive computing," *International Journal of Cognitive Informatics and Natural Intelligence* (IJCINI), vol. 6, no. 4, pp. 1-21, Oct-Dec 2012.

[6] Mauro Barni, Franco Bartolini, and Alessandro Piva, "Improved wavelet-based watermarking through pixel-wise masking," *IEEE Trans. Image Processing*, vol. 10, no. 5, pp. 783-791, May 2001.

[7] Prayoth Kumsawat, Kitti Attakitmongcol, and Arthit Srikaew, "A new approach for optimization in image watermarking by using genetic algorithms," *IEEE Trans. Sig. Processing*, vol.53, no.12, Dec. 2005.

[8] Chamidu Atupelage and Koichi Harada, "Perceptible content retrieval in DCE domain and semi-fragile watermarking technique for perceptible content authentication," *WSEAS Trans. Signal Processing*, vol. 4, no. 11, pp. 627–636, Nov. 2008.

[9] Pao-Ta Yu, Hung-Hsu Tsai, and Jyh-Shyan Lin, "Digital watermarking based on neural networks for color images," *Signal Processing*, vol. 81, no. 3, pp. 663-671, March 2001.

[10] Chun-Hsien Chou and Kuo-cheng Liu, "A perceptually tuned watermarking scheme for color images," *IEEE Trans. Image Processing*, vol. 19, no. 11, pp. 2966-2982, Nov. 2010.

[11] P. Tamije Selvy, V. Palanisamy, and S. Elakkiya, "A novel watermarking of images based on wavelet based contourlet transform energized by biometrics," *WSEAS Trans. Computers*, vol. 12, no. 3, pp. 105–115, March 2013.

[12] Hieu V. Dang and Witold Kinsner, "An intelligent digital color image watermarking approach based on wavelet transform and general regression neural network," in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics and Cognitive Computing*, ICCI*CC 2012, (Kyoto, Japan; August 22-24, 2012), pp. 115-123, 2012.

[13] Xiang-Yang Wang, Hong-Ying Yang, and Chang-Ying Cui, "An SVM-based robust digital image watermarking against desynchronization attacks," *Signal Processing*, vol. 88, no. 9, pp. 2193-2205, Sep. 2008.

[14] Hung-Hsu Tsai and Duen-Wu Sun, "Color image watermark extraction based on support vector machines," *Information Sciences*, vol. 177, no. 2, pp.550-569, Jan. 2007.

[15] Rui-min Shen, Yong-gang Fu, and Hong-tao Lu, "A novel image watermarking scheme based on support vector regression," *The Journal of Systems and Software*, vol. 78, no. 1, pp. 1-8, Oct. 2005.

[16] Chin-Shiuh Shieh, Hsiang-Cheh Huang, Feng-Hsing Wang, and Jeng-Shyang Pan, "Genetic watermarking based on transform-domain techniques," *Pattern Recognition*, vol. 37, no. 3, pp.555-565, March 2004.

[17] Prayoth Kumsawat, Kitti Attakitmongcol, and Arthit Srikaew, "An optimal robust digital image watermarking based on genetic algorithms in multiwavelet domain," *WSEAS Trans. Signal Processing*, vol. 5, no. 1, pp. 42–51, Jan. 2009.

[18] K. Ramanjaneyulu and K. Rajarajeswari, "Wavelet-based oblivious image watermarking scheme using genetic algorithm," *IET Image Process.*, vol. 6, no. 4, pp. 364-373, June 2012.

[19] Simon Haykin, *Neural networks: A comprehensive foundation*, Second Edi. Pearson, 1999, 823 pp.

[20] Donald F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, no. 6, pp. 568-576, Nov. 1991.

[21] Hieu V. Dang and Witold Kinsner, "A perceptual data hiding mathematical model for color image protection," *Journal of Advanced Mathematics and Applications*, vol. 1, no. 2, pp. 218–233, Dec. 2012

[22] I. J. Cox, M. Miller, and A. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127–1141, July 1999.

[23] P. Moulin and R. Koetter, "Data-hiding codes," *Proceedings of the IEEE*, vol. 93, no. 12, pp. 2083–2126, Dec. 2005.

[24] C. Cachin, "An information-theoretic model for steganography," *Information and Computation*, vol. 192, no. 1, pp. 41 – 56, 2004.

[25] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," *Information Theory, IEEE Transactions on*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[26] R. R. Yager, "On the theory of bags," *International Journal of General Systems*, vol. 13, no. 1, pp. 23–37, 1986.

[27] Witold Kinsner, *Switching Automata Theory*, Lecture Notes, Winnipeg, MB: University of Manitoba, 2012, 930 pp.

[28] F. A. P. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding-a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, July 1999.

[29] L. Marquez and T. Hill, "Function approximation using backpropagation and general regression neural networks," in *Proceedings of the 16th IEEE Int. Conf. System Sciences*, Hawaii, pp. 607-615, 1993.

[30] Chaofeng Li, Alan C. Bovik, and Xiaojun Wu, "Blind image quality assessment using a general regression neural network," *IEEE. Trans. Neural Netw.*, vol. 22, no. 5, pp. 793-799, May 2011.

[31] John Y. Goulermas, Panos Liatsis, Xiao-Jun Zeng, and Phil Cook, "Density-driven generalized regression neural networks (DD-GRNN) for function approximation." *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1683-1696, Nov. 2007.

[32] Ioannis Anagnostopoulos, Christos Anagnostopoulos, George Kouzas, and Dimitrios D. Vergados, "A generalised regression algorithm for web page categorisation," *Neural Comput. Appl.*, vol. 13, no. 3, pp. 229-263, Sep. 2004.

[33] Natalio Krasnogor and Jim Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474-488, Oct. 2005.

[34] K.C.Tan, E.F.Khor, and T.H.Lee, *Multiobjective evolutionary algorithms and applications*, London UK: Springer, 2010, 295 pp.

[35] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms," California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.

[36] Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Noritaka Tsukamoto, and Yusuke Nojima, "Implementation of multiobjective memetic algorithms for combinatorial optimization problems: A Knapsack problem case study," in *Multi-Objective Memetic Algorithms*, Chi-Keong Goh, Yew-Soon Ong, and Kay Chen Tan, (eds.) Berlin Springer, pp. 27-49, 2009.

[37] Ferrante Neri and Carlos Cotta, "Memetic algorithms and memetic computing optimization: a literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1-14, Feb. 2012.

[38] Xian Chen, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591-607, Oct. 2011.

[39] Christoph Bergmeir, Issac Triguero, Daniel Molina, Jose L. Aznarte, and Jose Manuel Benitez, "Time series modeling and forcasting using memetic algorithms for regime-switching models," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 23, no. 11, pp.1841-1847, Nov. 2012.

[40] David E. Goldberg, *Genetic algorithm in search, optimization, and machine learning*, Reading, MA: Addison-Wesley, 1989, 423p.

[41] kay C. Tan, Eik F. Khor, Tong H. Lee, and Ramasubramainan Sathikannan, "An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization," *Journal of Artificial Intelligence Research*, vol. 18, pp. 182-215, Feb. 2003.

[42] Daniel Molina, Manuel Lozano, Carlos Garcia-Martinez, and Francisco Herrera, "Memetic algorithms for continuous optimisation based on local search chains," *Evolutionary Computation*, vol. 18, no. 1, pp. 27-63, 2010.

[43] Andriana Lara, Gustavo Sanchez, Carlos A. Coello, and Oliver Schutze, "HCS: A new local serach strategy for memetic multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 112-132, Feb. 2010.

[44] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyaruvan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comp.*, vol. 6, no. 2, pp. 182-197, April 2002.

[45] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," Complex Syst., vol. 9, pp. 115-148, Nov. 1995.

[46] Andrzej Jaszkiewicz, "Gnetic local search for multi-objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1 pp.50-71, Feb. 2002.

[47] A. S. Lewis, and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Trans. Image Processing*, vol. 1, no. 2, pp. 244-250, April 1992.

[48] Raymond B. Wolfgang, Christine I. Podilchuk, and Edward J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE*, vol 87, no. 7, pp. 1108-1126, Jul. 1999.

[49] Yicong Zhou, Sos Agaian, Valencia M. Joyner, and Karen Panetta, "Two Fibonacci p-code based image scrambling algorithms," *SPIE Proceedings: Image processing - algorithms and systems VI*, vol. 6812, Paper #6812-15, San Jose, CA, January 2008.

[50] Fred Glover, "Tabu search: a tutorial," Interfaces, vol. 20, no. 4, pp. 74-94, Aug. 1990.

[51] M. Kutter, F. Jordan, and F. Bossen, "Digital watermarking of color images using amplitude modulation," *Joural of Electronic Imaging*, vol. 7, no. 2, pp. 326-332, 1998.

[52] David L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613-627, May 1995.

[53] Witold Kinsner, "Compression and its metrics for multimedia," in *Proc. of the 11th IEEE Intern. Conf. on Cognitive Informatics*, ICCI 02, (Calgary, Canada; August 19-20, 2002), 2002.