Egyptian Informatics Journal (2015) 16, 83–102

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

A novel gray-scale image watermarking using hybrid Fuzzy-BPN architecture

Charu Agarwal [a,*], Anurag Mishra [b], Arpita Sharma [c]

[a] Department of Computer Science, University of Delhi, Delhi, India
[b] Department of Electronics, Deendayal Upadhyay College, University of Delhi, Delhi, India
[c] Department of Computer Science, Deendayal Upadhyay College, University of Delhi, Delhi, India

Received 12 October 2014; revised 17 December 2014; accepted 12 January 2015
Available online 26 February 2015

KEYWORDS

Image watermarking;
Human visual system;
Fuzzy-BP network;
Hybrid intelligent system;
Normalized cross-correlation

Abstract In this paper, we model HVS characteristics using hybrid Fuzzy-BPN to implement a novel image watermarking scheme to embed a permuted binary watermark in gray-scale images. The signed images yield high values of full reference metrics – PSNR and SSIM which suggest their good visual quality. Extracted watermarks yield high normalized correlation values which indicate successful watermark recovery. Eight different image processing attacks are carried out to examine the robustness of embedding scheme. High computed values of normalized correlation from the attacked images clearly indicate that the proposed algorithm is robust against the selected attacks. Time complexity analysis indicates fast execution of watermarking processes. It is found that the proposed watermarking scheme is fast enough to carry out these operations on a real timescale. Overall, it is concluded that the Fuzzy-BPN is successful candidate for implementing novel gray-scale image watermarking scheme meeting real timelines.

1. Introduction

The digital watermarking of images has acquired an important dimension in the research of image processing applications. This is the process by which logo or watermark is embedded into images, giving those objects a unique digital identity that can be used for a variety of valuable applications. Traditionally, image watermarking is used to establish content authentication and ownership verification [1–3]. Watermarking techniques are categorized into two groups: spatial domain and frequency domain. In the first case, the pixel values of the cover image are directly altered by inserting the watermark. It leads to ease of implementation and low cost of operation but are generally not robust to affine transformations and image processing attacks [4,5]. In contrast, methods of second category transform the image into the frequency domain and then modify its coefficients to embed the watermark. This leads to robust watermark embedding. There are many transform

[*] Corresponding author. Tel.: +91 9891994775.
E-mail addresses: agarwalcharu2@rediffmail.com (C. Agarwal), anurag_cse2003@yahoo.com (A. Mishra), arpt1_mishra1@yahoo.com (A. Sharma).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2015.01.002
1110-8665 © 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

domain watermarking techniques such as discrete cosine transforms (DCT) [6–8], singular value decomposition (SVD) [9–11] and discrete wavelet transform (DWT) [12–16]. It is well known that human eyes are more sensitive to low frequency and midfrequency band coefficients [1,4]. Therefore, transform domain techniques are found to work well if the watermarks are embedded within the low frequency coefficients of the image. Moreover, it has been reported that among the transform domain methods, DWT is more suitable for achieving robust watermarking and imperceptible leading to good visual quality signed image [1,14].

Several research groups are working to optimize a generic algorithm which can be effectively used by the industry. On the other hand, the digital content, especially images and videos are subject to vulnerable attacks. The attacker may want to eliminate watermark from the signed image. Due to this reason, the embedding algorithm must be robust against common image processing attacks.

In view of this, image watermarking is presently perceived as an optimization problem which should be capable to balance out the requirements of imperceptibility and robustness. To this end, various pure and hybrid soft computing techniques such as Artificial Neural Networks (ANNs) [17–19], Fuzzy Inference Systems (FIS) [20,21], and Genetic Algorithms (GAs) [22] are used to embed and extract the digital content (watermark) from the given images. Besides this, various evolutionary algorithms are also used for this purpose. These are Fuzzy based Bacterial Foraging [23], Particle Swarm Optimization (PSO) [24] and Ant Colony Optimization (ACO) [25]. The objective of using these schemes is to implement a comprehensive watermarking scheme capable of balancing out the requirements of imperceptibility and robustness. This is particularly true as these two criteria are often found to be mutually exclusive to each other. The third important parameter required to be examined for information security – the watermark embedding capacity is assumed to be constant. This is because, unlike steganography, the size of the watermark to be embedded is very small as compared to that of the host image. Although, all these techniques have successfully carried out watermarking schemes, a comparison of these algorithms is important to understand the inherent strengths and weakness of individual methods. A brief survey of the research work conducted in this area is presented below.

Huang et al. [17] have proposed a novel blind watermarking technique based on BP neural network in wavelet domain. In this paper, with the aid of HVS characteristics, a scrambled watermark is embedded robustly and imperceptibly. They successfully apply a neural network to memorize the relation between the watermark and corresponding watermarked image. This way, the authors are successful to blindly recover the exact watermark from the signed image. Their experimental results show that the proposed scheme has a good imperceptibility and high robustness to various image processing attacks.

Mohanty et al. [26] have analyzed the implementation of the HVS model in the context of watermarking of images. For this purpose, they consider three HVS features namely: edge blocks of the image to be watermarked, the effect of variance across the blocks available in the host image (blocks having low variance are more sensitive to noise and the blocks having high variance are less sensitive to noise) and the computed values of the block intensity. They argue that the HVS based watermarking is expected to give good quality imperceptibly signed images.

Motwani et al. [20] have implemented a MAMDANI type Fuzzy Inference System (FIS) which uses as its input the HVS characteristics namely brightness, texture and edge sensitivities of the gray-scale image in question. The output of this inference system is successfully used to embed the watermark in the host image in the DWT domain. This FIS uses a set of 27 inference rules which are primarily based on the following facts:

(1) The eye is less sensitive to noise in those areas of the image where brightness is high or low.
(2) The eye is less sensitive to noise in highly textured areas, but among these, more sensitive near the edges.
(3) The eye is less sensitive in the regions with high brightness and changes in very dark regions.

The authors claim that the fuzzy based watermarking scheme is robust to image processing attacks and at the same time achieve a high level of imperceptibility.

Melin et al. [27] stressed upon the design of hybrid systems in general and its applications in pattern recognition and intelligent manufacturing in particular. The authors specifically highlight the importance of designing and implementing hybrid systems for real world applications. They combine several soft computing methodologies to build powerful hybrid intelligent systems that can fully exploit the advantages that individual technique offers. For this purpose, they consider face recognition, fingerprint recognition and voice identification as thrust areas. They amalgamate neural networks and fuzzy logic to examine face recognition in their paper. The authors also stress upon the use of Genetic Algorithms (GAs) to optimize the architecture of the face recognition system as well as the use of Neuro-Fuzzy system for fingerprint recognition. For designing of voice recognition system, they suggest a mixture of all three – Neural Networks, Fuzzy logic, and GA. They argue that the main disadvantage of the fuzzy system is their lack of adaptability to changing situations. For this reason, they suggest combining fuzzy logic with ANNs or GAs because ANNs and GAs give necessary adaptability to fuzzy systems.

Latif [28] proposed an adaptive digital image watermarking technique using fuzzy logic and tabu search. They had used Hadamard transform to transfer the image from spatial domain to transform domain. The selected transform includes some parameters that can be handled to control the requirements of watermarking such as robustness and imperceptibility. They had applied the transform parameters to enhance the robustness by tabu search and after embedding, the watermark is adapted to the image by exploiting human visual system characteristics using fuzzy gradient to ensure the imperceptibility. They claim that their experimental results have high imperceptibility as well as high robustness against variety of attacks.

Jacobsen [29] has first classified state-of-the-art intelligent systems utilized for various applications. He suggests integration of individual techniques by hybridization or fusion. This is aimed at overcoming the limitation of individual techniques. First, he categorizes the intelligent techniques into four categories (1) simple component system, (2) fusion based system, (3) hierarchical system and (4) hybrid system. Second, he introduces a unifying paradigm derived from concepts well known in artificial intelligence and agent community as a conceptual framework to better understand, modularize, compare and evaluate individual intelligent approaches. According to the author, neural networks are like a black box and hence it is not interpretable. Fuzzy techniques, on the other hand, show complementary behavior. He, thus, suggests integrating ANNs and Fuzzy system to overcome the limitation of the individual categories of intelligent systems.

Negnevitsky [30] proposed a novel design of Neuro-fuzzy hybrid system with heterogeneous and homogeneous structures. The system with a heterogeneous structure is used to diagnose myocardial perfusion from cardiac images which is subsequently used to predict cardiac attack. The system with a homogenous structure is developed for predicting an aircraft's trajectory during its landing abroad an aircraft carrier. According to the authors, both heterogeneous and homogeneous systems are quite successful in their implementation. The former is capable to predict a heart attack using the output of the fuzzy system to which the inputs are supplied by the trained ANN. The latter system using homogeneous capabilities is capable to predict the trajectory of the landing aircraft at two seconds in advance, based on the aircraft's current position. According to the author, although the field of hybrid intelligent systems is still evolving, and most hybrid tools are not yet particularly effective, neuro-fuzzy systems have already matured as an advanced technology with numerous successful applications. While neural networks can learn from data, the key benefit of fuzzy logic lies in its ability to model decision-making of humans.

Abraham [31] presented hybrid architecture of intelligent systems involving fuzzy-clustering algorithms, neural network learning, fuzzy inference systems and finally the evolutionary computation. He uses the concept of hierarchical layers to demonstrate the evolution of intelligence in his hybrid intelligent models. He argued that for two applications (approximating the expert behavior of multinational subsidiaries and web usage mining) the hybrid-architecture, as described above, produced better results than individual approaches in terms of low RMSE and high Cross-Correlation (CC). According to the author, better results are achieved by taking into account less number of learning rules. He claims that his approach is very suitable to hardware implementation.

1.1. Motivation

It can be inferred from this discussion that pure soft computing techniques have given solutions to a wide range of problems including gray-scale image watermarking. However, these techniques have their own advantages and disadvantages. For example, in case of an ANN, the precision is often limited to the least squares errors, the training time is quite large, the training data are sufficiently large and they have to be chosen over the entire range where the variables are expected to change. On the other hand, although the Fuzzy logic addresses the imprecision of inputs and outputs defined by fuzzy sets and allows greater flexibility in formulating a detail system description, yet it lacks in adaptability. It is, therefore, advisable to integrate ANN with Fuzzy logic. The resultant Neuro-Fuzzy systems are expected to extend the capabilities of the systems beyond either of the two pure techniques as applied individually. The proposed watermarking scheme implements a hybrid Fuzzy-BPN system which maps the fuzzy inputs to crisp outputs without involving a large training data set. Therefore, the problem of lack of adaptability of a pure fuzzy rule based system is also expected to be resolved by using it.

1.2. Research focus and contribution

This research work focuses on optimizing the trade-off between the twin parameters of image watermarking: imperceptibility and robustness. We, thus, propose a novel grayscale image watermarking scheme using the hybrid Fuzzy-BPN architecture developed by Lee and Lu [32] by taking into account the HVS characteristics of the gray-scale host images. To the best of our information, this hybrid network has not been used earlier for developing any image watermarking application. We employ three different characteristics of the Human Visual System (HVS) to embed and extract the watermark from five different gray-scale host images of size 256×256. These images are Lena, Baboon, Boat, Pepper and Man. The proposed applied work assumes more significance particularly because hybrid intelligent architectures are rarely in use for development of image watermarking schemes. Moreover, the processing time of the processes involved is also computed and discussed. It has been found that these processes do not consume much time to carry out computations. The HVS characteristics – luminance sensitivity, contrast sensitivity and edge sensitivity are fed to a Fuzzy-BPN as inputs. The Fuzzy-BPN is driven by the same set of 27 inference rules as proposed by Motwani et al. [20]. This network produces a weighting factor as its output. The weighting factor is used to embed a permuted binary watermark in the LL3 coefficients of the host image. The watermark is of size 32×32 pixels. In addition to this, we carry out eight different image processing operations over signed images as attacks to examine the robustness of the embedding scheme. These attacks are described in detail in Section 3.7. Perceptible quality of the watermarked and attacked images is quantified by PSNR and SSIM. The robustness of the embedding scheme is evaluated by Normalized Correlation, $NC(X, X^*)$. It is found that the embedding and extraction processes are well optimized and the hybrid embedding scheme is robust enough against the selected attacks.

The paper is organized as follows. Section 2 deals with theory of hybrid Fuzzy expert system based BPN proposed by Lee and Lu [32]. Section 3 described experimental details including embedding, extraction and robustness. Experimental results are discussed in Section 4 which finally concluded in Section 5. The list of references is given at the end.

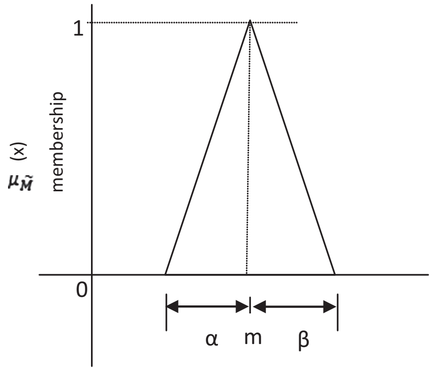


Figure 1 A triangular LR-type fuzzy number (m, α, β).

2. Mathematical formulations

This paper implements an Fuzzy-ANN based hybrid architecture, namely Lee and Lu's Fuzzy-BP network [32] for image watermarking. Fuzzy-BPN is a hybrid architecture which performs nonlinear mapping between fuzzy input vectors and crisp outputs. Therefore, it has the ability to process fuzzy numbers. The fuzzy numbers are represented in LR-type to reduce network complexity. The connection weights and biases are represented as fuzzy numbers to enhance fuzzy inference ability of this network. In addition, it uses a fuzzy neuron which performs fuzzy weighted summation, defuzzification and nonlinear mapping.

2.1. LR-type fuzzy number

The LR-type fuzzy number is a special representation of a fuzzy numbers in terms of that the fuzzy number $(\widetilde{M})$ can be triangularly expressed as $(m, \alpha, \beta)_{LR}$ as shown in Fig. 1. In this representation, m is the mean value of $\widetilde{M}$, α and β are left and right spreads of $\widetilde{M}$ respectively. If both α and β are zero, the LR-type fuzzy number indicates a crisp value.

2.2. Fuzzy neuron

The fuzzy neuron is the basic element of the Fuzzy BPN model. Fig. 2 illustrates the architecture of the fuzzy neuron. It performs nonlinear mapping between the weighted summation of fuzzy input vectors and crisp outputs. Given the fuzzy input vector $\widetilde{I} = (\widetilde{I}_0, \widetilde{I}_1, \ldots, \widetilde{I}_l)$ and fuzzy weight vector, $\widetilde{W} = (\widetilde{W}_0, \widetilde{W}_1, \ldots, \widetilde{W}_l)$, the fuzzy neuron computes the crisp output Oj.

In the above figure $\widetilde{net}_j$ is the fuzzy weighted summation and NET is the inference result computed as $NET = CE(\widetilde{net}_j)$. The function CE is the centroid operation of the triangular fuzzy number and is treated as a defuzzification operation which maps fuzzy weighted summation value to a crisp output value. In the present case, the function f is sigmoid function which performs nonlinear mapping between the input and output.

2.3. Fuzzy BPN architecture

Fuzzy-BPN is a three layer feed forward architecture. The three layers are as follows: input layer, hidden layer, and output layer. As in case of BPN, the execution of Fuzzy-BPN proceeds in two stages namely,

1. Learning or Training.
2. Inference.

Fig. 3 illustrates a l–m–n (l input neurons, m hidden neurons, and n output neurons) architecture for the Fuzzy-BP network. Let $\widetilde{I}_p = (\widetilde{I}_{p1}, \widetilde{I}_{p2}, \ldots, \widetilde{I}_{pl})$ be the input pattern string supplied to the Fuzzy-BPN architecture shown in Fig. 3. In this notation, $p = 1, 2, 3, \ldots, N$ being the pth pattern for a total of N input patterns that the Fuzzy-BPN needs to be trained. The bias value is taken as $I_0 = (1, 0, 0)$. $\widetilde{I}_{pi}$ is the ith input component of the pth input pattern and is an LR-type triangular fuzzy number.

Let $\widetilde{O}_{pi}$ be the output value of the ith input neuron, O'_{pj} and O''_{pk} are jth and kth crisp defuzzified outputs of the hidden and output layer neurons respectively. $\widetilde{W}_{ji}$ is the fuzzy weight between ith input neuron and jth hidden neuron. $\widetilde{W}_{kj}$ is the fuzzy weight between jth hidden neuron and kth output neuron.

2.4. Algorithm for training of Fuzzy-BPN

Listing 1 illustrates the algorithm for training of Fuzzy-BPN.

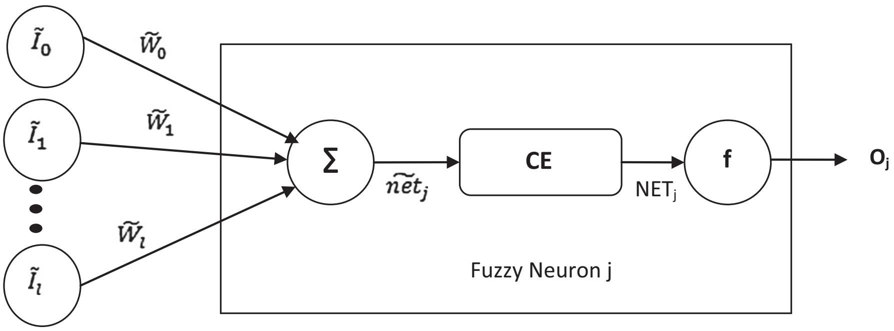


Figure 2 The architecture of jth fuzzy neuron.

Listing 1: Algorithm Fuzzy_BP_TRAINING

```
/* let the configuration of Fuzzy-BPN be $l$–$m$–$n$ */
Step 1:   Randomly generate the initial weight sets $\widetilde{W}$ for the
          input–hidden layer where each
          $\widetilde{W}_{ji} = (W_{mji}, W_{\alpha ji}, W_{\beta ji})$ is an LR_Type fuzzy number.
          Also generate the weight set $\widetilde{W}'$ for hidden–output
          layer where $\widetilde{W}'_{kj} = \left(W'_{mkj}, W'_{\alpha kj}, W'_{\beta kj}\right)$
Step 2:   Let $(\widetilde{I}_p, D_p), \forall p = 1, 2, \ldots, N$ be $N$ input–output
          pattern sets that Fuzzy-BPN needs to be trained with.
          Here, $\widetilde{I}_p = (\widetilde{I}_{p0}, \widetilde{I}_{p1}, \ldots, \widetilde{I}_{pl})$ where each $\widetilde{I}_{pi}$ is an LR-
          type fuzzy number, i.e. $\widetilde{I}_{pi} = (\widetilde{I}_{pmi}, \widetilde{I}_{p\alpha i}, \widetilde{I}_{p\beta i}) \cdot D_p$ is a
          crisp output
Step 3:   Let ITR is a variable which denotes the number of
          iterations. Set the counter for the number of iterations
          to zero and the counter for number of pattern sets to
          be trained to one. i.e. COUNT_OF_ITR = 0; and
          $p = 1$;
Step 4:   Assign values to training parameters – learning rate ($\eta$)
          and a constant value ($\gamma$). Initialize the variables used to
          compute change in the weights for input–hidden and
          hidden–output layers at time $t-1$ respectively as:
          $\Delta\widetilde{W}(t-1) = 0$;
          $\Delta\widetilde{W}'(t-1) = 0$;
          The weights at time $t-1$ for input–hidden and
          hidden–output layer are given respectively as
          $\widetilde{W}(t-1) = 0$;
          $\widetilde{W}'(t-1) = 0$;
Step 5:   Get next pattern set $(\widetilde{I}_p, D_p)$
          For the input neuron, assign
          $\widetilde{O}_{pi} = \widetilde{I}_{pi}, \forall i = 1, 2, \ldots, l$; and $\widetilde{O}_0 = (1, 0, 0)$
Step 6:   For hidden neurons compute
          $O'_{pj} = f(NET_{pj}), \forall j = 1, 2, \ldots, m$; and $O'_{p0} = 1$
              where $NET_{pj} = CE(\sum_{i=0}^{l} \widetilde{W}_{ji}\widetilde{O}_{pi})$
Step 7:   For output neurons compute
                  $O''_{pk} = f(NET'_{pk}), \forall k = 0, 1, \ldots, n-1$
              where $NET'_{pk} = CE\left(\sum_{j=0}^{m} \widetilde{W}_{kj}O'_{pj}\right)$
Step 8:   Compute change of weights $\Delta\widetilde{W}'(t)$ for hidden–output
          layer as follows:
                  $\Delta\widetilde{W}'(t) = -\eta\nabla E_p(t) + \gamma\Delta\widetilde{W}'(t-1)$
Step 9:   Compute change of weights $\Delta\widetilde{W}(t)$ for input- hidden
          layer as follows:
                  $\Delta\widetilde{W}'(t) = -\eta\nabla E_p(t) + \gamma\Delta\widetilde{W}'(t)$
Step 10:  Update weights by using:
                $\widetilde{W}(t) = \widetilde{W}(t-1) + \Delta\widetilde{W}(t)$ for input–hidden
          layer and
             $\widetilde{W}'(t) = \widetilde{W}'(t-1) + \Delta\widetilde{W}'(t)$ for hidden–output
          layer.
Step 11:  $p = p + 1$;
          if ($p \leqslant N$) goto step 5;
Step 12:  COUNT_OF_ITR = COUNT_OF_ITR + 1
          if COUNT_OF_ITR < ITR
          {Reset pointer to first pattern in the training set;
          $p = 1$;
          goto step 5;
          }
Step 13:  Output $\widetilde{W}$ and $\widetilde{W}'$ as the final weight sets
```

2.5. Inference by Fuzzy-BPN

Once the Fuzzy-BPN is trained for a given set of input–output patterns a definite number of times, it is ready for inference. Given a set of patterns $\widetilde{F}_p$ to be inferred, where $\widetilde{F}_p = (\widetilde{F}_{p1}, \widetilde{F}_{p2}, \ldots, \widetilde{F}_{pl})$ and $\widetilde{F}_{pi}$ is an LR-type fuzzy number given by $\widetilde{F}_{pi} = (\widetilde{F}_{pmi}, \widetilde{F}_{p\alpha i}, \widetilde{F}_{p\beta i})$. The aim is to obtain O_p, the output corresponding to $\widetilde{F}_p$. O_p is computed in one pass by allowing $\widetilde{F}_p$ to pass through the series of computations. The term O''_k computed by the output neurons is the output corresponding to $\widetilde{F}_p$.

Listing 2 illustrates the algorithm for Fuzzy-BPN inference.

Listing 2: Algorithm Fuzzy-BPN Inference

```
/* Input:
$\widetilde{F}_p, \forall p = 1, 2, \ldots, M$ and
$\widetilde{W}$ and $\widetilde{W}'$ be the weight sets obtained after training fuzzy BP
*/
Step 1:     $p = 1$
Step 2:     Get next pattern $\widetilde{F}_p$
Step 3:     For input neurons compute $\widetilde{O}_0 = (1, 0, 0)$
Step 4:     For hidden neurons compute
            $O'_{p0} = 1$
            $O'_{pj} = f(NET_{pj}) \quad j = 1, 2, \ldots, m$;
            where $NET_{pj} = CE\left(\sum_{i=0}^{l} \widetilde{W}_{ji}\widetilde{O}_{pi}\right)$
Step5:      For output neurons compute
            $O'_{pk} = f(NET'_{pk}), \forall k = 0, 1, \ldots, n-1$
            where $NET'_{pk} = CE\left(\sum_{j=0}^{m} W_{kj}O'_{pj}\right)$
Step 6:     Obtain the associated output
            $O'_{pk}, \forall k = 0, 1, 2, \ldots, n-1$
Step 7:     $p = p + 1$
            If ($p \leqslant M$) goto step 2
            else
            Stop
```

3. Classification of simulation work

The present problem is classified for simulation under following categories.

3.1. Block coding and computing HVS parameters of original image

The host image (I) of size 256 × 256 is first divided into blocks of size 8 × 8 pixel each in spatial domain. The obtained 1024 blocks are then transformed into transform domain using DCT method. Three HVS characteristics namely: luminance sensitivity, contrast sensitivity and edge sensitivity are computed over these blocks using Eqs. (1)–(3).

3.1.1. The luminance (brightness) sensitivity

The DC coefficients of the DCT blocks of the host image are used as luminance sensitivity according to the formula given by Eq. (1)

$$L_i = \frac{X_{DC,i}}{X_{DCM}} \tag{1}$$

where $X_{DC,i}$ denotes the DC coefficient of the ith block and X_{DCM} is the mean value of the DC coefficients of all the blocks put together.

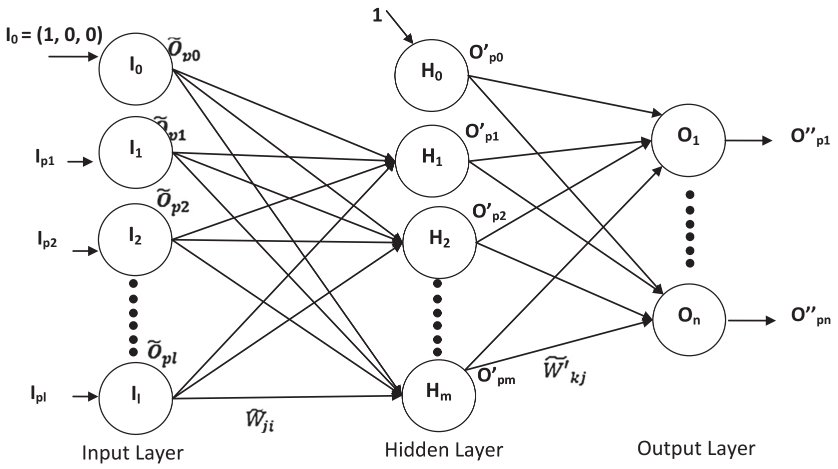


Figure 3 An architecture of three-layered Fuzzy BPN.

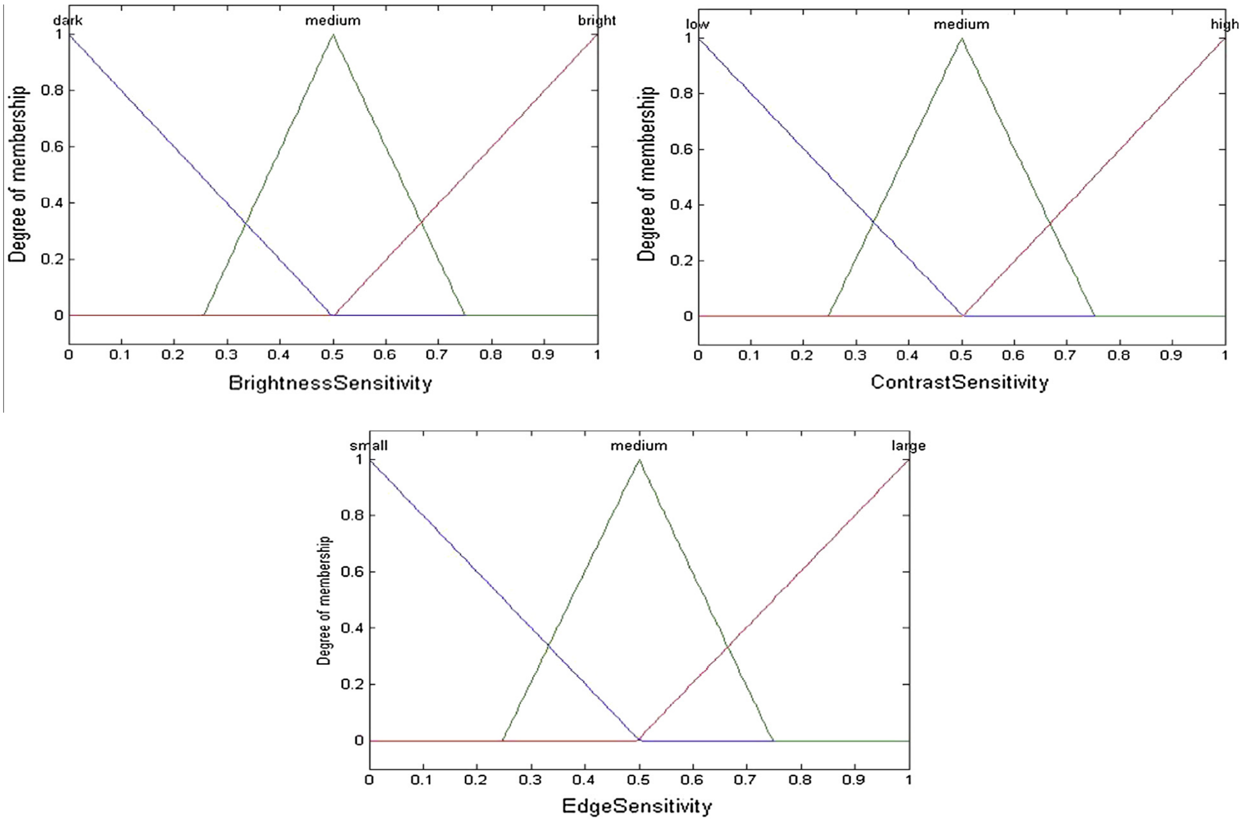


Figure 4 Fuzzy membership functions used for three input attributes – brightness sensitivity, contrast sensitivity and edge sensitivity.

3.1.2. The contrast sensitivity

The texture of a region (block) in an image can be quantified by its contrast sensitivity. The computed variance value of an image block is the direct metric to quantify this parameter. For this purpose, a Matlab routine proposed by Gonzalez et al. [33] is used. The implementation of this routine is given by Eq. (2).

$$t = \mathrm{statxture}(f) \tag{2}$$

where *f* is the input image or the sub-image (block) and *t* is the 7 – element row vector. These elements are (1) average gray level, (2) average contrast, (3) smoothness measure, (4) third moment, (5) measurement of uniformity, (6) entropy and (7) normalized variance value.

3.1.3. The edge sensitivity

As the edge is detected in the image using threshold operation, edge sensitivity can be quantified as a natural corollary to the computation of the block threshold T. The Matlab image processing toolbox implements graythresh() routine which computes the block threshold using histogram-based Otsu's method [33]. The implementation of this routine is given by Eq. (3)

$$T = \text{graythresh}(f) \tag{3}$$

where f is the host sub-image (block) in question and T is the computed threshold value. These parameters are also calculated by other researchers [20,26]. They, however, use different formulations to compute them.

3.2. Fuzzy linguistic terms for three input attributes

Fig. 4 illustrates the fuzzy linguistic terms associated with luminance (brightness) sensitivity, contrast sensitivity and edge sensitivity computed. Note that each linguistic variable consists of three fuzzy sets. For example, luminance sensitivity has dark, medium and bright levels. Contrast sensitivity has low, medium and high levels and edge sensitivity has small, medium and large levels. This is done to decompose these parameters into fuzzy equivalent variables to constitute the fuzzy inference rules. These fuzzy sets are represented in LR-type. Table 1 illustrates the LR-type fuzzy number equivalents for the associated attribute values.

3.3. Fuzzy inference rules

Fuzzy-BPN is driven by the set of 27 inference rules as listed below:

Rules	Luminance sensitivity	Contrast sensitivity	Edge sensitivity	Weight (output)
1	Dark	Low	Small	Least
2	Dark	Medium	Small	Least
3	Dark	High	Small	Least
4	Medium	Low	Small	Least
5	Medium	Medium	Small	Least
6	Medium	High	Small	Least
7	Bright	Low	Small	Least
8	Bright	Medium	Small	Least
9	Bright	High	Small	Least
10	Dark	Low	Medium	Less
11	Dark	Medium	Medium	Higher
12	Dark	High	Medium	Higher
13	Medium	Low	Medium	Less
14	Medium	Medium	Medium	Average
15	Medium	High	Medium	Average
16	Bright	Low	Medium	Less
17	Bright	Medium	Medium	Average
18	Bright	High	Medium	Higher
19	Dark	Low	Large	Less
20	Dark	Medium	Large	Higher
21	Dark	High	Large	Highest
22	Medium	Low	Large	Less
23	Medium	Medium	Large	Average
24	Medium	High	Large	Higher
25	Bright	Low	Large	Less
26	Bright	Medium	Large	Higher
27	Bright	High	Large	Highest

Table 1 LR-type fuzzy number equivalents for associated attributes.

	Fuzzy Set	LR-type fuzzy number
Brightness Sensitivity	Dark	(0, 0.001, 0.5)
	Medium	(0.5, 0.25, 0.25)
	Bright	(1, 0.5, 0.0001)
Contrast Sensitivity	Low	(0, 0.001, 0.5)
	Medium	(0.5, 0.25, 0.25)
	High	(1, 0.5, 0.0001)
Edge Sensitivity	Small	(0, 0.001, 0.5)
	Medium	(0.5, 0.25, 0.25)
	Large	(1, 0.5, 0.0001)

The final outcome of application of these rules is the suitable output of the expert system and is given by one of the five crisp output values namely: Least (0.0), Less (0.25), Average (0.5), Higher (0.75) and Highest (1.0).

3.4. Training and inference of Fuzzy-BPN for gray-scale image watermarking

In the present work, a three layered Fuzzy-BPN with a 3–3–1 configuration (3 input neurons, 3 hidden neurons and 1 output neuron) is used. This network is trained using Lee and Lu's [32] training procedure. The values of learning rate (η) and constant (γ) are 0.9 and 0.1 respectively. To train the network, 27 HVS rules are used as given in Section 3.3. In accordance with Table 1, the computed LR-type fuzzy numbers are used to train the network for a definite number of iterations. In this simulation, the iteration number is optimized to 50 iterations. During trials of this algorithm, we observe that beyond ITR = 50, the visual quality and robustness of signed images do not improve. Hence, the embedding is optimized for ITR = 50. After training, let W and W' be the weight sets obtained for the input–hidden and hidden–output layers respectively. These weight sets are used in Fuzzy-BPN inference.

For the inference, we first compute brightness sensitivity, contrast sensitivity and edge sensitivity of all blocks of the host image using Eqs. (1)–(3). Thereafter, the block wise computed values are converted into their equivalent LR-type fuzzy numbers using Table 1. The set of LR-type fuzzy numbers of three HVS parameters is inferred by the Fuzzy-BPN using W and W' weight sets, which results in a crisp output O''. Therefore, for each block, O'' is computed in one pass by allowing each set of three parameters (block wise) to pass through the series of computations of Fuzzy-BP network. The crisp outputs are further used in watermark embedding using a prespecified formula given by Eq. (4).

3.5. Embedding the watermark

In the present simulation, we use five gray-scale host images represented by (I) to demonstrate watermark embedding. These images are Lena, Baboon, Boat, Man and Pepper. The watermark (W) embedded in I is a permuted binary image of size $m \times m$ pixels. Fig. 5 depicts the block diagram of the proposed watermark embedding scheme. The formula for embed-

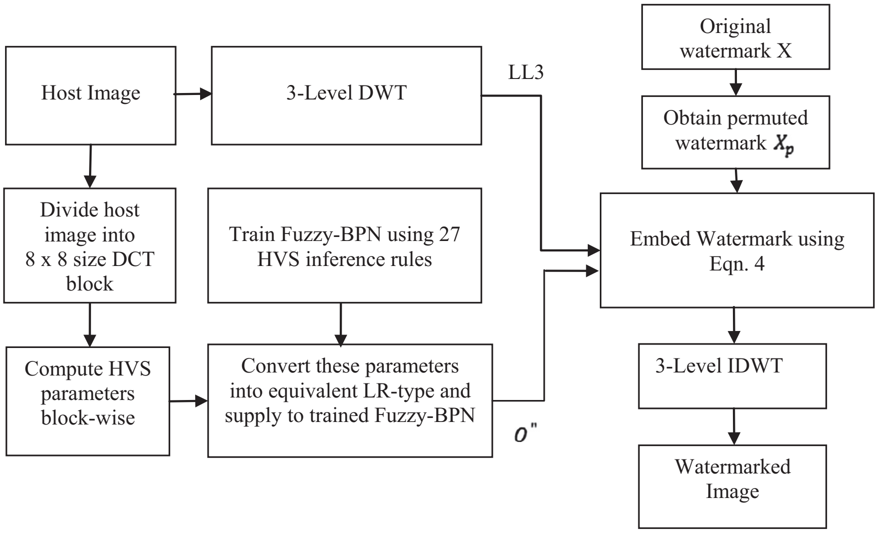

Host Image
3-Level DWT
LL3
Original watermark X
Obtain permuted watermark X_p
Divide host image into 8 x 8 size DCT block
Train Fuzzy-BPN using 27 HVS inference rules
Embed Watermark using Eqn. 4
Compute HVS parameters block-wise
Convert these parameters into equivalent LR-type and supply to trained Fuzzy-BPN
O"
3-Level IDWT
Watermarked Image

Figure 5 Block diagram of watermark embedding scheme.

ding the watermark used in the present work is given by Eq. (4) [1]

$$LL3' = LL3 * ((k * O'' * X_p) + 1) \tag{4}$$

where $LL3$ is the 3 – level DWT low frequency region of the host image, O'' is the crisp output of Fuzzy-BPN, X_p is the original permuted watermark, k is the watermark scaling coefficient and $LL3'$ is the DWT low frequency region of the signed image. The scaling coefficient k is optimized to be 0.07 for the binary watermark. The optimization of k is explained in Section 4. The watermark embedding procedure is given by Listing 3.

Quality assessment of the signed images (I') is done by computing two full reference quality assessment metrics PSNR and SSIM given by Eqs. (5) and (6) respectively.

$$PSNR = 10\,log_{10}\left(\frac{I_{max}^2}{MSE}\right) \tag{5}$$

where I_{max} is the maximum possible pixel value of the image I and MSE is the mean square error.

$$SSIM = \frac{(2\mu_I\mu_{I'} + C_1)(2\sigma_{II'} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_I^2 + \sigma_{I'}^2 + C_2)} \tag{6}$$

where $\mu_I = \frac{1}{n}\sum_{i=1}^{n} I_i$ and $\mu_{I'} = \frac{1}{n}\sum_{i=1}^{n} I'_i$ are mean intensity or luminance component of image signals x and y respectively. $C_1 = (J_1L)^2$ and $C_2 = (J_2L)^2$ are constants with L being the dynamic range of the grayscale image (0–255) and $J_1 \ll 1$ and $J_2 \ll 1$ being small constants. Besides this,

$$\sigma_I^2 = \frac{1}{n-1}\sum_{i=1}^{n}(I_i - \mu_I)^2, \sigma_{I'}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(I'_i - \mu_{I'})^2 \text{ and } \sigma_{II'}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(I_i - \mu_I)(I'_i - \mu_{I'})$$

where σ_I and $\sigma_{I'}$ are signal contrast given by standard deviation for I and I' respectively and is used to estimate contrast comparison for SSIM. Listing 3 gives watermark embedding algorithm.

Listing 3: Watermark Embedding Algorithm

Step 1: Divide host image into 8×8 size blocks in spatial domain and compute DCT of all blocks
Step 2: Compute luminance sensitivity, contrast sensitivity and edge sensitivity of all blocks of the host image using Eqs. (1)–(3) respectively
Step 3: Train the Fuzzy-BP network using 27 Fuzzy BP inference rules derived from HVS model and retain W and W' weight sets
Step 4: Convert block wise above computed three parameters into their equivalent LR type fuzzy numbers and supply them as input to the trained Fuzzy-BPN to obtain the crisp output (O'')
Step 5: Obtain permuted watermark X_p by performing pseudo-random permutation on original watermark X
Step 6: Apply three-level DWT on the original image to obtain the subband LL3
Step 7: Embed the watermark using the formula given in Eq. (4)
Step 8: Compute three-level IDWT to obtain watermarked (signed) image

3.6. Watermark extraction from signed image and computation of normalized cross-correlation parameter NC(X, X)*

The extraction procedure is inverse of that of embedding and is informed in the present work. Fig. 6 depicts the block diagram of the watermark extraction scheme. The formula for extracting the watermark used in the present work is given in Eq. (7) [1]

$$X_p^* = (LL3'' - LL3)/(k * wO'') \tag{7}$$

where LL3 is the low frequency DWT coefficient of the host image, wO'' is the crisp output of Fuzzy-BPN, X_p^* is the extracted permuted watermark, k is the watermark scaling coefficient and $LL3''$ is the DWT coefficient of the signed image. The scaling coefficient k is optimized to be 0.07 for the binary watermark. The watermark extraction procedure is given by Listing 4.

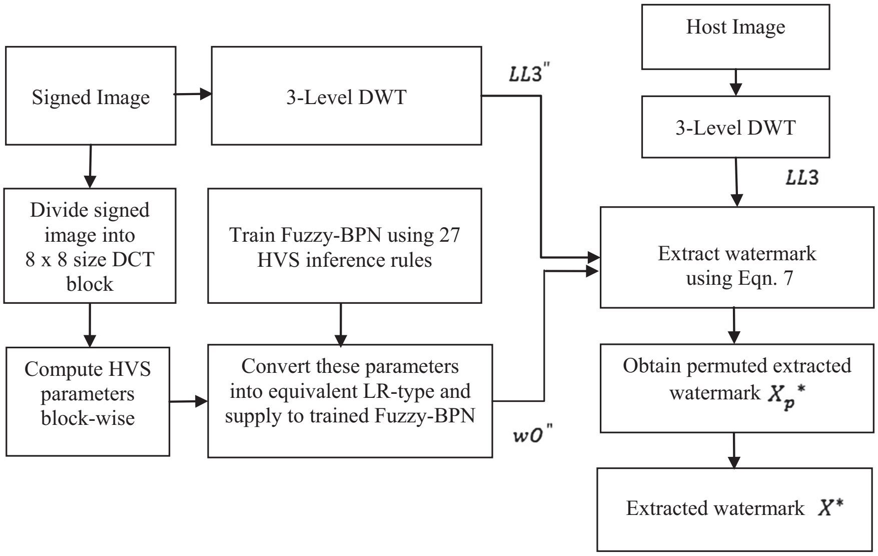

Signed Image
3-Level DWT
LL3''
Host Image
3-Level DWT
LL3
Divide signed image into 8 x 8 size DCT block
Train Fuzzy-BPN using 27 HVS inference rules
Extract watermark using Eqn. 7
Compute HVS parameters block-wise
Convert these parameters into equivalent LR-type and supply to trained Fuzzy-BPN
wO''
Obtain permuted extracted watermark $X_p{}^*$
Extracted watermark X^*

Figure 6 Block diagram of watermark extraction scheme.

Listing 4: Watermark Extraction Algorithm

```
Step 1: Divide the signed image into $8 \times 8$ size blocks in spatial
        domain and compute DCT of all blocks
Step 2: Compute luminance sensitivity, variance (contrast
        sensitivity) and threshold (edge sensitivity) of all blocks of
        the host image using Eqs. (1)–(3) respectively
Step 3: Train the Fuzzy-BP network using 27 Fuzzy BP inference
        rules derived from HVS model and retain $W$s and $W'$s
        weight sets
Step 4: Convert block wise above computed three parameters into
        their equivalent LR type fuzzy numbers and supply them
        as input to the trained Fuzzy-BPN to obtain the crisp
        output ($wO''$)
Step 5: Compute three level DWT of original and signed images
        and obtain LL3 and LL3'' subbands respectively for the
        two images
Step 6: Subtract the computed coefficients of original image from
        those of signed image using the formula given by Eq. (7)
Step 7: Reconstruct the extracted watermark $X^*$ from the
        computed permuted extracted watermark $X_p^*$
```

The embedded and recovered watermarks X and X^* respectively are compared and correlated by using the correlation parameter $NC(X, X^*)$ given by Eq. (8).

$$NC(X, X^*) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[X(i,j) \cdot X^*(i,j)]}{\sum_{i=1}^{m}\sum_{j=1}^{n}[X(i,j)]^2} \tag{8}$$

3.7. Robustness studies

To examine the issue of robustness of the proposed embedding scheme, the watermarked images are subject to eight different image processing attacks. These are

(1) JPEG compression (Q = 90, 75, 50, 25 and 10).
(2) Rotation (180°).
(3) Gaussian Blur (radius = 1.0 unit).
(4) Gaussian Noise addition (5% and 10%).
(5) Median Filter (aperture = 3.0 and 5.0).
(6) Wiener Filter (aperture = 3.0).
(7) Scaling ($1 \to 2 \to 1$).
(8) Cropping: (a) upper left 1/4 of the watermarked image is cropped and the removed portion is filled with original image, (b) left half of the watermarked image is cropped and the removed portion is filled with original image, (c) upper left 1/4 of the watermarked image is cropped and the removed portion is filled with 1s, (d) left half of the watermarked image is cropped and the removed portion is filled with 1s, (e) left half of the watermarked image is cropped and the removed portion is filled with 0s.

As mentioned in Section 1 that the signed as well as the attacked images are examined for their visual quality by using two full reference metrics PSNR and SSIM.

4. Results and discussion

This section presents results obtained by carrying out the embedding and extraction of watermark into five standard gray-scale host images. These images are Lena, Baboon, Boat, Man and Pepper. These processes are explained below:

4.1. Embedding and extraction operations

Fig. 7(a–e) depicts five standard gray-scale host images – Lena, Baboon, Boat, Man and Pepper of size 256×256. Fig. 7(f) depicts original binary watermark of size 32×32. Fig. 8(a–e) depicts the signed images obtained by embedding the same binary watermark in host images of Fig. 7(a–e) respectively. The computed values of PSNR and SSIM are mentioned on top of these signed images. These values indicate that their visual quality is good. Fig. 9(a–e) depicts watermarks recovered from signed images of Fig. 8(a–e) respectively. The $NC(X, X^*)$ values are mentioned on top of these watermarks. High computed $NC(X, X^*)$ values indicate that extraction is quite successful.

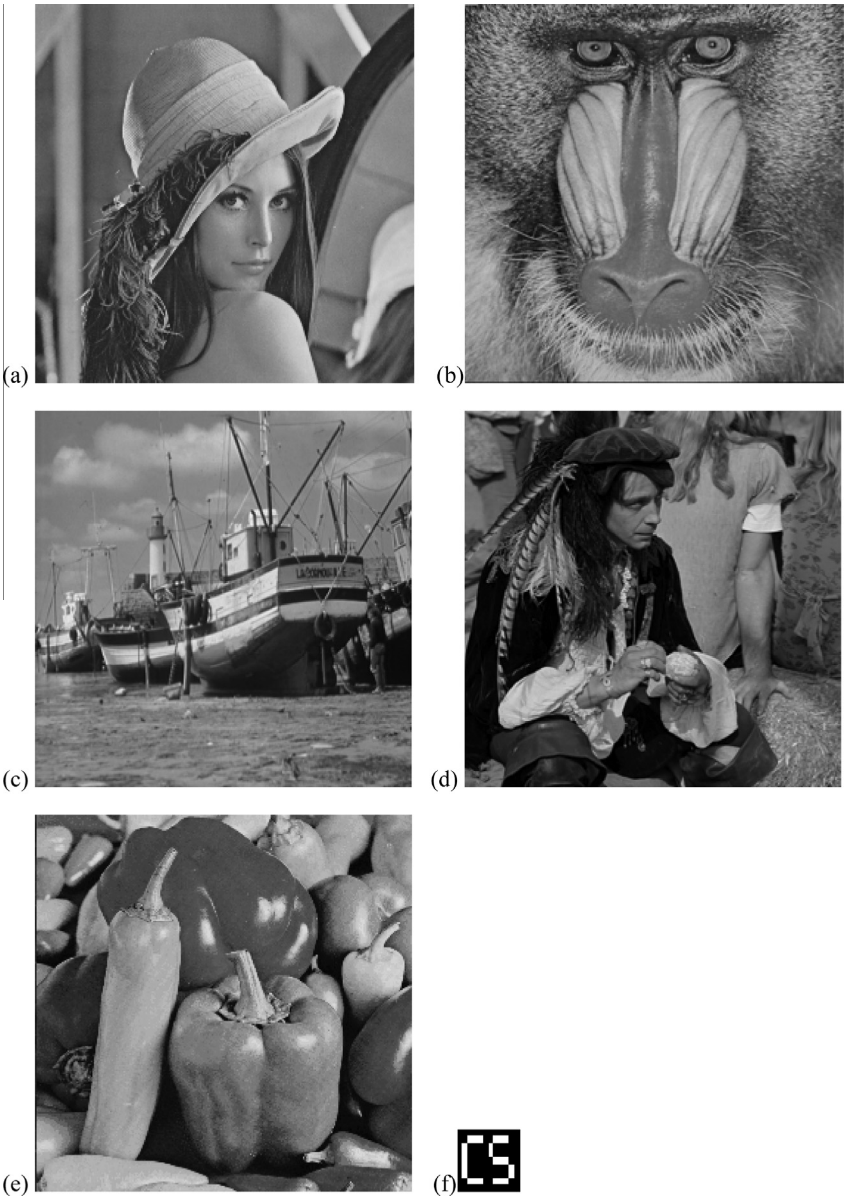


Figure 7 (a) Lena.bmp, (b) Baboon.bmp, (c) Boat.bmp, (d) Man.bmp, (e) Pepper.bmp and (f) Binary Watermark.

4.2. Executing image processing operations

To examine the robustness of the proposed embedding scheme, eight different image processing operations are executed on all five signed images of Fig. 8(a–e). These attacks are namely: (1) JPEG compression (Q = 90, 75, 50, 25 and 10), (2) Rotation (180°), (3) Gaussian Blur (radius = 1.0 unit), (4) Gaussian Noise addition (5% and 10%), (5) Median Filter (aperture = 3.0 and 5.0), (6) Wiener Filter (aperture = 3.0), (7) Scaling ($1 \rightarrow 2 \rightarrow 1$), and (8) Cropping ((a) upper left 1/4 of the watermarked image is cropped and the removed portion is filled with original image, (b) left half of the watermarked image is cropped and the removed portion is filled with original image, (c) upper left 1/4 of the watermarked image is cropped and the removed portion is filled with 1s, (d) left half of the watermarked image is cropped and the removed portion is filled with 1s, (e) left half of the watermarked image is cropped and the removed portion is filled with 0s). Table 2 compiles the computation results obtained from signed images after executing the said attacks.

As far as intelligent computing techniques are concerned, ANNs, fuzzy rule based methods, machine learning techniques, evolutionary algorithms have been frequently used to implement digital image watermarking in gray-scale and colored images. We have discussed in Section 1, the limitations of these pure techniques to carry out this simulation. For example, ANNs suffer from time complexity issues while Fuzzy Inference System (FIS) based methods suffer from the adaptability issue. To overcome this limitation, hybrid intelligent techniques are proposed. Neuro-Fuzzy architecture is one such example which has been successfully used in the area of image processing in general [29]. As far as digital watermarking is concerned, these hybrid techniques have been used very recently.

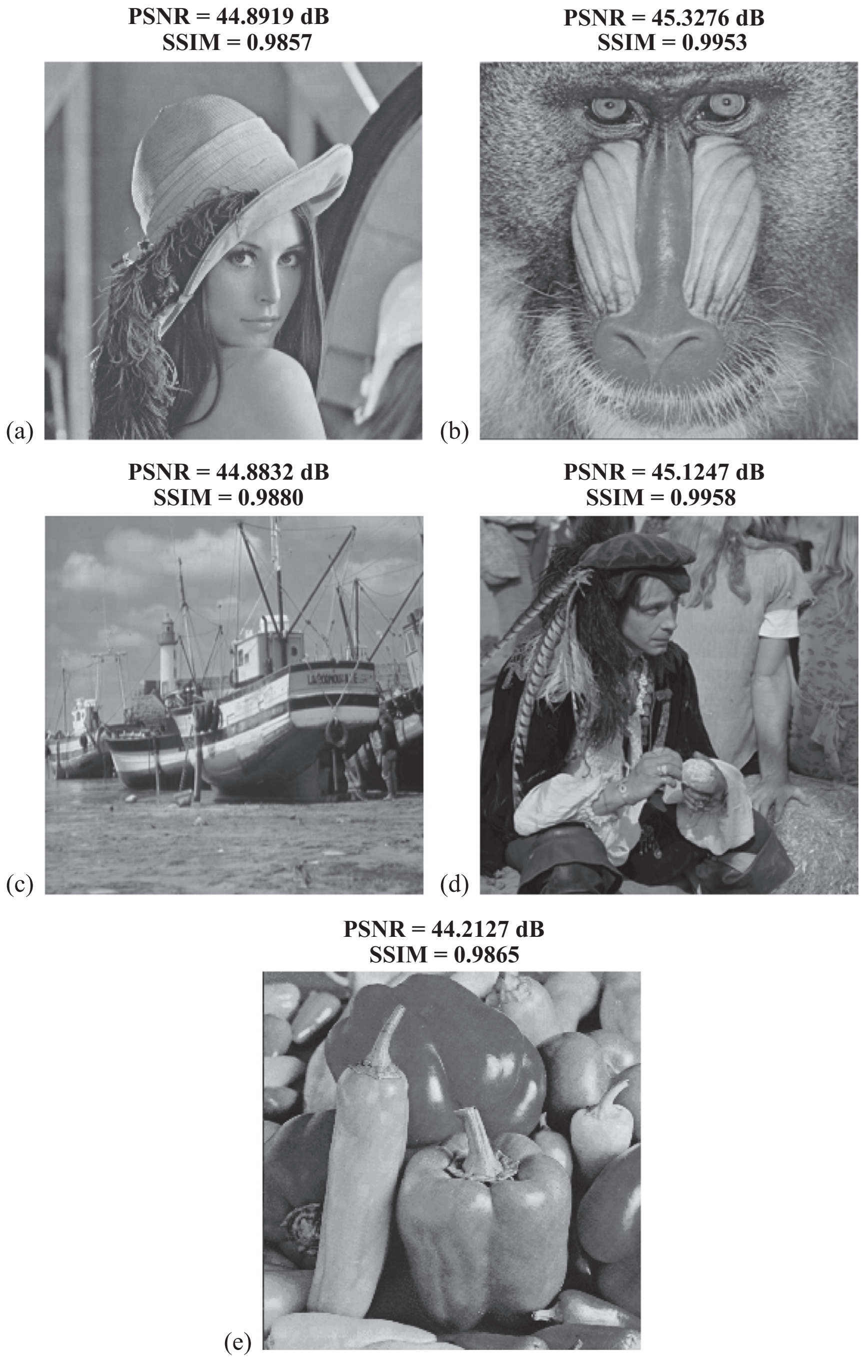


Figure 8 Signed Images: (a) Lena, (b) Baboon, (c) Boat, (d) Man and (e) Pepper.

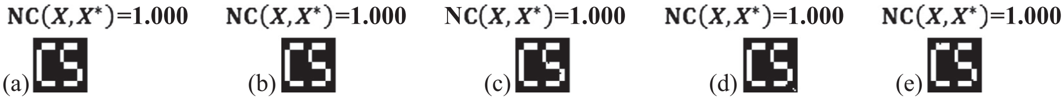


Figure 9 (a–e) Extracted watermarks and their respective $NC(X, X^*)$ values obtained from Fig. 6(a–e).

A careful observation of these results indicates the following points:

(i) A Neuro-Fuzzy architecture given by Lee and Lu is successfully used for the first time to implement a novel digital image watermarking scheme in five different grayscale images.

(ii) High computed values of PSNR and SSIM indicate that signed images have good imperceptibility.

(iii) High computed values of $NC(X, X^*)$ indicate that watermark recovery is successful and efficient. Note that NC value less than 0.5, in case of JPEG (Q = 10), 10% Gaussian Noise and Cropping half of the watermarked image and filling the missing portion with zeros and ones

Table 2 PSNR, SSIM and $NC(X, X^*)$ values for attacked Lena, Baboon, Boat, Man and Pepper images.

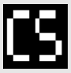
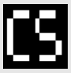
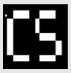
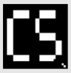
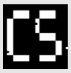
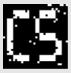
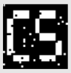
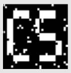
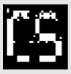
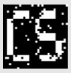
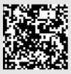
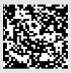
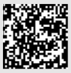
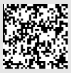
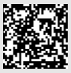
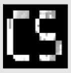
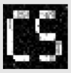
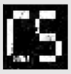
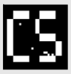
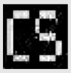
Attack	Image	PSNR (dB)	SSIM	$NC(X, X^*)$	Extracted watermark
JPEG ($Q = 90$)	Lena	37.7256	0.9771	1	
	Boat	36.3090	0.9740	1	
	Baboon	38.0097	0.9882	1	
	Man	38.0168	0.9801	1	
	Pepper	35.9240	0.9794	1	
JPEG ($Q = 75$)	Lena	33.7320	0.9468	1	
	Boat	30.7580	0.9534	1	
	Baboon	34.0424	0.9570	0.9707	
	Man	34.2261	0.9522	0.9754	
	Pepper	32.2892	0.9429	0.9871	
JPEG ($Q = 50$)	Lena	32.0685	0.9235	0.9931	
	Boat	28.8021	0.9250	0.9957	
	Baboon	32.1784	0.9312	0.9352	
	Man	31.5007	0.9305	0.9317	
	Pepper	30.9166	0.9147	0.9262	

Table 2 (*continued*)

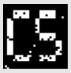
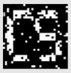
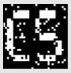
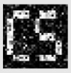
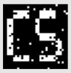
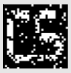
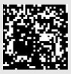
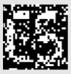
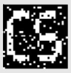
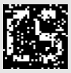
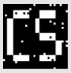
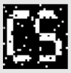
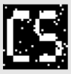
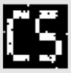
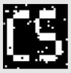
Attack	Image	PSNR (dB)	SSIM	$NC(X, X^*)$	Extracted watermark
JPEG (Q = 25)	Lena	30.5849	0.8495	0.9156	
	Boat	27.2534	0.8353	0.9174	
	Baboon	30.4060	0.8756	0.8476	
	Man	29.2871	0.8733	0.8353	
	Pepper	29.7018	0.8215	0.8327	
JPEG (Q = 10)	Lena	27.5762	0.7421	0.4024	
	Boat	24.4228	0.6780	0.4057	
	Baboon	26.7240	0.7465	0.4102	
	Man	26.2753	0.7628	0.4367	
	Pepper	26.8561	0.7241	0.4356	
Rotation (180°)	Lena	30.5623	0.8634	0.9445	
	Boat	29.6718	0.8217	0.9442	
	Baboon	31.7856	0.9023	0.9435	
	Man	30.7877	0.8725	0.9441	
	Pepper	31.9826	0.9147	0.9411	

(*contined on next page*)

Table 2 (*continued*)

Attack	Image	PSNR (dB)	SSIM	$NC(X, X^*)$	Extracted watermark
Gaussian Blur (radius = 1.0 unit)	Lena	29.3344	0.8608	0.8777	
	Boat	25.8000	0.7457	0.9045	
	Baboon	28.1718	0.8534	0.8798	
	Man	27.6478	0.8656	0.8872	
	Pepper	28.1756	0.8407	0.8744	
5% Gaussian noise	Lena	24.2990	0.5861	0.7186	
	Boat	24.1734	0.7359	0.7207	
	Baboon	24.2065	0.7385	0.6867	
	Man	24.1609	0.7350	0.6632	
	Pepper	24.4171	0.6451	0.7409	
10% Gaussian noise	Lena	21.3786	0.4521	0.4653	
	Boat	21.2172	0.5955	0.4972	
	Baboon	21.3296	0.4993	0.4689	
	Man	21.0164	0.5987	0.4994	
	Pepper	21.7215	0.5175	0.4950	

Table 2 (*continued*)

Attack	Image	PSNR (dB)	SSIM	$NC(X, X^*)$	Extracted watermark
Median filter (aperture = 3.0)	Lena	30.6025	0.8741	0.9713	
	Boat	28.5706	0.7505	0.9514	
	Baboon	30.7565	0.8746	0.9460	
	Man	28.2310	0.8210	0.9564	
	Pepper	30.1147	0.8404	0.9670	
Median filter (aperture = 5.0)	Lena	27.7746	0.8004	0.9141	
	Boat	23.6360	0.5560	0.8995	
	Baboon	28.0053	0.7926	0.9027	
	Man	27.3212	0.7628	0.9121	
	Pepper	27.2637	0.7778	0.9055	
Weiner filter (aperture = 3.0)	Lena	34.0646	0.8990	0.9416	
	Boat	27.1040	0.7893	0.9377	
	Baboon	32.3936	0.8959	0.9309	
	Man	31.3489	0.8901	0.9321	
	Pepper	32.4523	0.8624	0.9437	

(*contined on next page*)

Table 2 (*continued*)

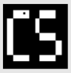
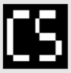
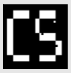
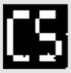
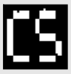
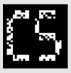
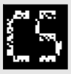
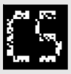
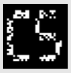
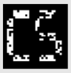
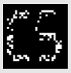
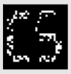
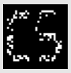
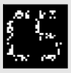
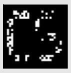
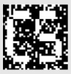
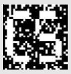
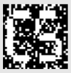
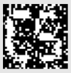
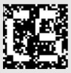
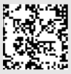
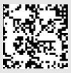
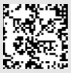
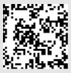
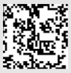
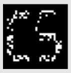
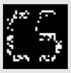
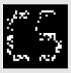
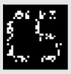
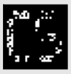
Attack	Image	PSNR (dB)	SSIM	$NC(X, X^{*})$	Extracted watermark
Scaling (resized to half and then restored to original size)	Lena	37.3142	0.9686	1	
	Boat	34.3891	0.9722	1	
	Baboon	37.6178	0.9781	0.9957	
	Man	36.5624	0.9639	0.9927	
	Pepper	35.7387	0.9636	0.9957	
Crop (quarter of the watermarked image and fill the missing portion with host image)	Lena	43.9137	0.9894	0.9734	
	Boat	43.6023	0.9961	0.9756	
	Baboon	44.8849	0.9970	0.9723	
	Man	43.9265	0.9848	0.9741	
	Pepper	44.8433	0.9897	0.9730	
Crop (half of the watermarked image and fill the missing portion with host image)	Lena	46.5371	0.9951	0.9143	
	Boat	45.6212	0.9975	0.9130	
	Baboon	46.3081	0.9981	0.9159	
	Man	45.1976	0.9916	0.9104	
	Pepper	42.1840	0.9932	0.9154	

Table 2 (*continued*)

Attack	Image	PSNR (dB)	SSIM	$NC(X, X^*)$	Extracted watermark
Crop (quarter of the watermarked image and fill the missing portion with 1's)	Lena	11.6860	0.7945	0.5495	
	Boat	11.2043	0.7514	0.5495	
	Baboon	13.7472	0.8292	0.5474	
	Man	12.1566	0.7825	0.5501	
	Pepper	10.9036	0.7650	0.5580	
Crop (half of the watermarked image and fill the missing portion with 1's)	Lena	8.8502	0.6230	0.3776	
	Boat	8.5834	0.5234	0.3776	
	Baboon	9.9621	0.6240	0.3756	
	Man	8.2010	0.5662	0.3743	
	Pepper	8.3594	0.5581	0.3785	
Crop (half of the watermarked image and fill the missing portion with 0's)	Lena	8.3483	0.4847	0.4923	
	Boat	8.6485	0.4868	0.4914	
	Baboon	8.6677	0.4914	0.4919	
	Man	8.6223	0.4811	0.4901	
	Pepper	8.4141	0.4825	0.4920	

attacks indicate difficult recover or possible nonrecovery of the watermark.

(iv) The plot of PSNR, SSIM and $NC(X, X^*)$ or signed and attacked Lena images with respect to k is shown in Fig. 10(a–c) respectively. According to Fig. 10(c), as $NC(X, X^*)$ values get saturated at around $k = 0.07$ and the respective PSNR and SSIM values tend to decrease beyond $k = 0.07$, we consider this value of k to be the optimized one for all our practical computations.

(v) Table 3 compiles and compares two different mathematical quantities – PSNR and $NC(X, X^*)$ for signed and attacked Lena image. The results compiled by us are compared with those proposed by Huang et al. [17] and Motwani et al. [20]. Both these groups consider different category of watermarks for their respective embedding/extraction schemes. Huang et al. [17] used a binary image of size 25 × 25 while Motwani et al. [20] used the weighting factor as watermark generated pixel wise by the Mamdani Fuzzy Inference System. The entry ‘NA’ in Table 3 indicates nonavailability of a specific attack in these papers. It is clear that our Fuzzy-BPN based hybrid watermarking scheme outperforms both other schemes for obtaining signed and attacked images except for the cropping attack.

(vi) Table 4 compiles computed time spans for embedding and extraction processes. Note that these computed time spans are of the order of few seconds only.

(vii) We compute time complexity for all five host images using a system with following specifications: Processor – Intel Core i3-3110M CPU @ 2.40 GHz with 2 GB RAM and Windows 8 (64-bit operating System). Other results are also computed by the same system. Note that the maximum time consumed to embed and extract the binary watermark is about 8.8 s per image. This indicates that the embedding and extraction algorithm is quite fast and the proposed hybrid intelligent scheme can be safely used for real time processing of images for watermarking purpose. Due to its faster processing, we intend to use this algorithm for video watermarking on a real timescale as well.

(viii) The comprehensive effects of image processing attacks indicate that the embedding algorithm is robust enough against the selected/used attacks. Results compiled in Table 2 show that the attacked images produce good PSNR, SSIM and $NC(X, X^*)$ values.

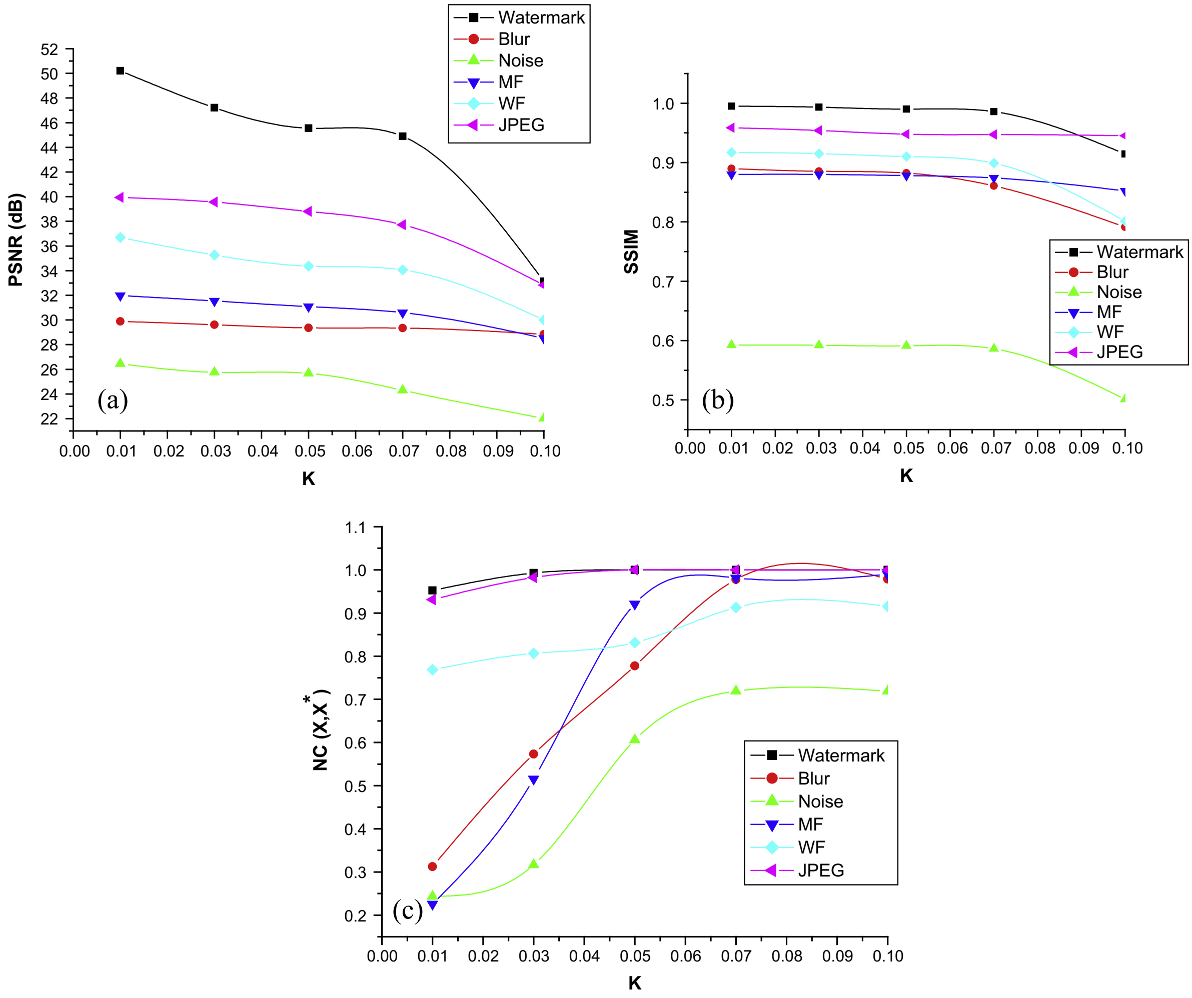


Figure 10 (a) Plot of PSNR vs. scaling parameter, (b) plot of SSIM parameter (k), and (c) plot of $NC(X, X^*)$ vs. scaling parameter (k).

Table 3 Comparison of $NC(X, X^*)$ for our method with Huang et al. [17] and Motwani et al. [20] for Lena.

	Huang et al. [17]	Motwani et al. [20]	Our method
PSNR (dB)	44.55	41.53	46.8919
Median filter (aperture = 5.0)	0.9063	NA	0.9144
Wiener filter (aperture = 3.0)	NA	0.88	0.9416
Rotation (180°)	NA	0.854	0.9445
Scaling (resized to half and then restored to original size)	0.7851	0.796	1.000
JPEG (Q = 90)	1.0000	NA	1.000
JPEG (Q = 75)	1.0000	NA	1.000
JPEG (Q = 50)	0.9154	NA	0.9931
JPEG (Q = 25)	0.8105	NA	0.9156
Cropping (half of the watermarked image and fill the missing portion with 0's)	0.9018	NA	0.5914

Table 4 Time (in seconds) taken by proposed algorithm in embedding and extraction.

Procedure	Lena	Baboon	Boat	Man	Pepper
Training of Fuzzy-BPN followed by Embedding	8.5781	7.7969	8.0625	7.0938	8.6094
Training of Fuzzy-BPN and Extraction from the signed image	8.5938	8.0313	8.1094	7.1126	8.7456
Training of Fuzzy-BPN and Extraction from the signed image after blur attack	8.1634	7.9567	8.0645	7.3413	8.2381
Training of Fuzzy-BPN and Extraction from signed image after 5% Gaussian noise attack	8.3477	7.8834	8.1267	7.5432	8.6735
Training of Fuzzy-BPN and Extraction from signed image after median filter (aperture = 3.0) attack	8.4256	7.9262	8.2435	7.6712	8.7827
Training of Fuzzy-BPN and Extraction from signed image after JPEG (Q = 90) attack	8.3423	7.9026	8.1523	7.7812	8.2983
Training of Fuzzy-BPN and Extraction from signed image after crop (quarter of the watermarked image and fill the missing portion with host image) attack	8.4422	7.9856	8.4875	7.4671	8.6528

(ix) As the proposed watermarking scheme is based on a Neuro-Fuzzy hybrid architecture, it is expected to resolve the problems which otherwise affect ANNs and Fuzzy Inference Systems (FIS) for the reasons mentioned in Section 1. The most important aspect of a commercial watermarking application is its requirement of real time implementation of successful embedding, extraction and meeting the robustness criteria. The Neuro-Fuzzy hybrid architecture based proposed watermarking scheme is quite successful to fulfill these criteria.

5. Conclusions

This paper presents a novel image watermarking technique which involves three basic characteristics of the HVS model namely – Luminance, Contrast Sensitivity computed using block variance and Edge sensitivity computed using block threshold value. These HVS characteristics are modeled using Lee and Lu's Fuzzy-BP network to implement watermarking in five gray-scale images – Lena, Baboon, Boat, Man and Pepper. Lee and Lu's Fuzzy-BPN is trained by 27 inference rules. For each block of the host image, the trained Fuzzy-BPN produces a single crisp output value which is used to embed a binary watermark of size 32×32, into the host image in the transform (DWT) domain.

The major contribution of the proposed scheme is the application of Fuzzy expert system based back-propagation network for gray-scale image watermarking. To examine the robustness of the proposed algorithm, eight different image processing attacks are executed over signed images. Experimental results show that the proposed scheme yields high values of PSNR and SSIM, which indicate that the signed and attacked images have good perceptible quality. The watermark is also extracted from the signed and attacked images using Fuzzy-BPN. The embedded and extracted watermarks are compared and $NC(X, X^*)$ parameter is computed. The $NC(X, X^*)$ values are found to be within expected range and well above the required threshold value which indicates that the embedding and extraction processes are well optimized with a good time complexity. Thus, the proposed algorithm is found to be extremely suitable for practical real time applications. The performance of the proposed scheme is compared with pure intelligent techniques [17,20] and it is shown that the hybrid intelligent technique based watermarking scheme outperforms the performance of two other schemes based on pure intelligent methods.

References

[1] Cox I, Kilian J, Leighton FT, Shamoon T. Secure spread spectrum watermarking for multimedia. IEEE Trans Image Process 1997;6:1673–87.

[2] Liu R, Tan T. A SVD-based watermarking scheme for protecting rightful ownership. IEEE Trans Multimedia 2002;4(1):121–8.

[3] Abdallah Emad E, Ben Hamza A, Bhattacharya Prabir. Improved image watermarking scheme using fast Hadamard and discrete wavelet transforms. J Electron Imag 2007;16(3):0330201–09.

[4] Nikolaidis N, Pitas I. Robust image watermarking in the spatial domain. Signal Process 1998;66(3):385–403.

[5] Liu J-C, Chen S-Y. Fast two-layer image watermarking without referring to the original image and watermark. Image Visual Comput 2001;19(14):1083–97.

[6] Briassouli A, Strintzis MG. Locally optimum nonlinearities for DCT watermark detection. IEEE Trans Image Process 2004;13(12):1604–17.

[7] Patra JC, Phua JE, Bornand C. A novel DCT domain CRT-based watermarking scheme for image authentication surviving JPEG compression. Digit Signal Process 2010;20:1597–611.

[8] Hernandez J, Amado M, Perez-Gonzalez F. DCT-domain watermarking techniques for still images: detector performance analysis and a new structure. IEEE Trans Image Process 2000;9(1):55–67.

[9] Jain C, Arora S, Panigrahi PK. A reliable SVD based watermarking scheme. adsabs.harvard.edu/abs/2008arXiv0808.0309J; 2008.

[10] Liu F, Liu Y. A watermarking algorithm for digital image based on DCT and SVD. In: IEEE congress on image and signal processing, Sanya, Hainan, China, vol. 1; 2008. p. 380–3.

[11] Liu R, Tan T. An SVD-based watermarking scheme for protecting rightful ownership. IEEE Trans Multimedia 2002;4:121–8.

[12] Xianghong T, Lu L, Lianjie Y, Yamei N. A digital watermarking scheme based on DWT and vector transform. In: Proceeding of international symposium on intelligent multimedia, video and speech processing; 2004. p. 635–8.

[13] Ohnishi J, Matsui K. Embedding a seal in to a picture under orthogonal wavelet transform. In: Proceedings of IEEE international conference on multimedia and computing system, Hiroshima, Japan; 1996. p. 514–21.

[14] Meerwald P, Uhl A. A survey on wavelet domain watermarking algorithms. In: Proceedings of SPIE, electronic imaging, security and watermarking of multimedia contents III, vol. 4314; 2001. p. 505–16.

[15] Barni M, Bartolini F, Piva A. Improved wavelet based watermarking through pixel wise masking. IEEE Trans Image Process 2001;10:783–91.

[16] Dawei Z, Guanrong C, Wenbo L. A chaos based robust wavelet domain watermarking algorithm. Chaos Solitons Fractals 2004;22:47–54.

[17] Huang Song, Zhang Wei, Feng Wei, Yang Huaqian. Blind watermarking scheme based on neural network. In: Seventh world congress on intelligent control and automation; 2008. p. 5985–9.

[18] Lou Der-Chyuan, Hu Ming-Chiang, Liu Jiang-Lung. Healthcare image watermarking scheme based on human visual model and back-propagation network. J CCIT 2008;37(1):151–62.

[19] Mei Shi-chun, Li Ren-hou, Dang Hong-mei, Wang Yun-kuan. Decision of image watermarking strength based on artificial-neural networks. In: Ninth international conference on neural information processing, vol. 5; 2008. p. 2430–4.

[20] Motwani Mukesh C, Harris Jr Fredrick C. Fuzzy perceptual watermarking for ownership verification. In: International conference on image processing, computer vision, and pattern recognition, Las Vegas, Nevada; 2009.

[21] Agarwal Charu, Mishra Anurag, Sharma Arpita. Digital image watermarking in DCT domain using fuzzy inference system. In: Twenty fourth IEEE Canadian conference on electrical and computer engineering (CCECE 2011); 2011. p. 822–5.

[22] Shieh C, Huang H, Wang F, Pan J. Genetic watermarking based on transform domain techniques. Pattern Recogn Lett 2004;37:555–65.

[23] Huang Hsiang-Cheh, Chen Yueh-Hong, Abraham Ajith. Optimized watermarking using swarm-based bacterial foraging. J Inf Hiding Multimedia Signal Process 2010;1(1):51–8.

[24] Tsai Hung-Hsu, Jhuang Yu-Jie, Lai Yen-Shou. An SVD-based image watermarking in wavelet domain using SVR and PSO. Appl Soft Comput 2012;12(8):2442–53.

[25] Loukhaoukha Khaled, Chouinard Jean-Yves, Taieb Mohamed Haj. Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization. J Inf Hiding Multimedia Signal Process 2011;2(4):303–19.

[26] Mohanty Saraju P, Ramakrishnan KR, Kankanhalli Mohan. A dual watermarking technique for images. ACM Multimedia 1999;Part 2:49–51.

[27] Melin P, Castillo O, Kacprzyk J, Pedrycz W. Design of hybrid intelligent systems. Fuzzy Information Processing Society; 2007, ISBN: 1-4244-1213-7.

[28] Latif Alimohammad. An adaptive digital image watermarking scheme using fuzzy logic and tabu search. J Inf Hiding Multimedia Signal Process 2013;4:250–71.

[29] Jacobsen Hans-Arno. A generic architecture for hybrid intelligent systems. In: IEEE world congress on computational intelligence, vol. 1; 1998. p. 709–14.

[30] Negnevitsky Michael. Hybrid neuro-fuzzy system: heterogeneous and homogeneous structures. In: World congress on computer science and information engineering; 2008. p. 533–40.

[31] Abraham Ajit. Hybrid intelligent systems: evolving intelligence in hierarchical layers. Stud-fuzz, vol. 173. Berlin Heidelberg: Springer-Verlag; 2005. p. 159–79.

[32] Lee Hahn Ming, Lu Bing Hui. Fuzzy BP: a neural network model with fuzzy inference. ICANN 1994:1583–8.

[33] Gonzalez Rafel C, Woods Richard E, Eddins Steven L. Digital image processing using MATLAB. Pearson Education; 2005. p. 406–7.