

## Research Article

# A New Robust Watermarking Scheme to Increase Image Security

**Hossein Rahmani, Reza Mortezaei, and Mohsen Ebrahimi Moghaddam**

*Electrical and Computer Engineering Department, Shahid Beheshti University, G.C., Tehran 1983963113, Iran*

Correspondence should be addressed to Mohsen Ebrahimi Moghaddam, m\_moghaddam@sbu.ac.ir

Received 12 December 2009; Revised 7 July 2010; Accepted 16 October 2010

Academic Editor: Yingzi Du

Copyright © 2010 Hossein Rahmani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In digital image watermarking, an image is embedded into a picture for a variety of purposes such as captioning and copyright protection. In this paper, a robust private watermarking scheme for embedding a gray-scale watermark is proposed. In the proposed method, the watermark and original image are processed by applying blockwise DCT. Also, a Dynamic Fuzzy Inference System (DFIS) is used to identify the best place for watermark insertion by approximating the relationship established between the properties of HVS model. In the insertion phase, the DC coefficients of the original image are modified according to DC value of watermark and output of Fuzzy System. In the experiment phase, the CheckMark (StirMark MATLAB) software was used to verify the method robustness by applying several conventional attacks on the watermarked image. The results showed that the proposed scheme provided high image quality while it was robust against various attacks, such as Compression, Filtering, additive Noise, Cropping, Scaling, Changing aspect ratio, Copy attack, and Composite attack in comparison with related methods.

## 1. Introduction

Owing to the recent advances in network and multimedia techniques, digital images may be transmitted over the nonsecure channels such as the Internet. Therefore, the enforcement of multimedia copyright protection has become an important issue in literature.

Watermarking and cryptography are two standard multimedia security methods. However, cryptography is not an effective method because it does not provide permanent protection for the multimedia content after delivery to consumers, because, after decryption there is no protection for the documents. Digital watermarking technologies allow users to hide appropriate information in the original image that is imperceptible during normal use but readable by special application. Therefore, the major purpose of digital watermarks is to provide protection for intellectual property that is in digital format. To evaluate a watermark system, the following attributes are generally considered [1, 2].

- (1) *Readability*. A watermark should convey as much information as possible, statistically detectable, enough to identify ownership and copyright unambiguously.

- (2) *Security*. Only authorized users gain access to the watermark data.
- (3) *Imperceptibility*. The embedding process should not introduce any perceptible artifacts into original image and not degrade the perceived quality of image.
- (4) *Robustness*. The watermark should be able to withstand various attacks while can be detected in the extraction process.

The most important watermarking schemes are invisible where are secure and robust. Moreover, in the invisible watermarking, the embedding locations are secret, and only the authorized persons who have the secret keys can extract the watermark.

On the other hand, the watermarking algorithms are classified also as: the methods which require the original information and secret keys for extracting watermark are called private watermark algorithms. The methods which require the watermark information and secret keys are called semiprivate or semiblind algorithms, and ones which need secret keys rather than the original information are called blind watermark algorithms.

In another classification, digital watermarking algorithms can be divided into two groups: spatial domain [5–7] and frequency domain [8–12] methods according to the processing domain of the host image. The spatial domain algorithms are simple and the watermark can be damaged easily, but the frequency domain algorithms can resist versus intensity attack and watermark information cannot be damaged easily [13].

However, in all frequency domain watermarking schemes, there is a conflict between robustness and transparency. If the watermark is embedded in the lower-frequency bands, the scheme would be robust to attacks but the watermark may be difficult to hide. On the other hand, if the watermark is embedded in the higher-frequency bands, it would be easier to hide the watermark but the scheme has less resistant to attacks. Therefore, finding a proper place to embed the watermark is very important.

In 1996, Cox et al. [14] advised that the watermark should be embedded in the low-frequency coefficients of DCT domain to ensure the robustness. To improve this method, Lu et al. [15] used a cocktail watermark to increase robustness and HVS to maintain high fidelity of the watermarked image. Barni and Hsu [16, 17], respectively, recommended that the watermark should be embedded in the middle frequency coefficients to reduce the distortion. But Huang et al. in [18] points out that the DC coefficient is more proper to be used for embedding watermark, and this conclusion is obtained based on his robustness test between the DC coefficient and two low-frequency coefficients.

Also, DWT as another frequency transform technique has been used by many researchers such as Xie and Arce for digital image watermarking [19]. The proposed method by Zhao et al. in [20] is a sample of DCT/DWT domain-based method which uses a dual watermarking scheme exploiting the orthogonality of image subspaces to provide robust authentication. As other examples, in [21, 22], the proposed DCT/DWT methods embed a binary visual watermark by modulating the middle-frequency components. These two methods are robust to common image attacks; but geometric attacks are still challenges. In [23], another approach to combine DWT and DCT has been proposed to improve the performance of the DWT-based watermarking algorithms. In this method, watermarking is done by embedding the watermark in the first and second level of DWT subbands of the host image, followed by the application of DCT on the selected DWT subbands. The combination of these two transforms improved the watermarking performance considerably in comparison with DWT-only watermarking approach.

Most of the existing watermarking methods use a pseudorandom sequence or binary image as a watermark. However, using grayscale images as watermarks has drawn much attention for copyright protection since many logos are grayscale in nature. One of the methods that hide a grayscale watermark image in original image was proposed by Mahanty and Bhargava [24]. In this method, at first, based on Human Visual System (HVS), the most perceptually important region of original image is found. Then, a compound watermark is created to insert in this region of the

original image. For creation of compound watermark, the synthetic image is created by Gaussian and Laplacian random number generator. The choice of these two distributions for modeling the DC and AC coefficients of image DCT is motivated by empirical results presented in Reininger and Gibson [25] and Mohanty et al. [26]. Next, the original watermark is embedded in insensitive area of synthetic image using any DCT-based visible watermarking algorithm. Asatryan proposed another method that combines spatial and frequency domain to hide a grayscale watermark in grayscale original image by mapping the values of DCT coefficients of compressed watermark image to the interval [0,255] (max and min value of grayscale image) by a fixed linear transform and inserts these values in the original image [27]. But, this method introduces perceptible artifacts into original image and degrades the perceived quality of image.

In this paper, we have proposed a new robust watermarking method in frequency domain to insert a gray level watermark in an image. The proposed method is more robust and makes image with higher quality than related ones. The basic idea of the proposed method is based on this fact that most of the signal energy of the DCT block is compacted in the DC component and the remaining energy is distributed reductively in the AC components in zigzag scan order [4]. Also, for most images, the main characteristics of the DCT coefficients in one block have high correlation with the adjacent blocks. Gonzales et al. [3] described a technique which estimates the first five AC coefficients precisely. In this method, DC values of a  $3 \times 3$  neighborhood of blocks are used to estimate the AC coefficients for the center block. They did not consider variations in the image in AC coefficients estimation, but Veeraswamy and Kumar in [4] proposed a new method that considered the variation in the image and accordingly AC coefficients have been estimated with different equations. This method is better than Gonzales method in terms of reduced blocking artifacts and improved PSNR value. Based on these ideas, here, at first, a grayscale watermark image is created by applying DCT on each  $b \times b$  nonoverlapping block of original grayscale watermark image and setting all AC coefficients of each one to zero. Then, the original image is divided into  $a \times a$  nonoverlapping blocks and DCT is applied on each  $a \times a$  block. Next, a Dynamic Fuzzy Inference System (DFIS) is used to select the number of original image blocks for embedding watermark. Finally, DC value of each  $b \times b$  DCT block of watermark image is embedded in DC value of  $a \times a$  DCT block of original image by using the output of the DFIS. In the extraction process, DCT is applied on the test image to extract the DC coefficients of each  $b \times b$  DCT block of watermark and the AC coefficients of each  $b \times b$  DCT block of extracted watermark are estimated based on proposed technique by Veeraswamy and Kumar [4] to construct the watermark with higher quality. The proposed method was tested on several bench mark images using StirMarkMATLAB software and its results were satisfactory. The results showed that the proposed method created the high-quality watermarked images while they were more robust against attacks such as JPEG compression, additive noise, filtering, cropping.



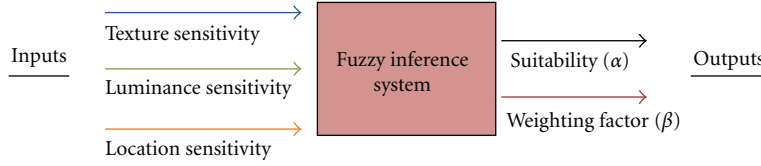


FIGURE 1: Inputs and outputs of Dynamic Fuzzy Inference System (DFIS).

The rest of paper has been organized as follows: In Section 2, the proposed approach has been introduced and in Section 3, the proposed method has been motivated and structurally compared with related ones. Section 4 describes the experimental results and in Section 5, the paper has been concluded.

## 2. Proposed Algorithm

In this section, the proposed algorithm is described in detail. The algorithm is divided into four parts: block selection, watermark creation, watermark embedding and watermark extraction, which are described in Sections 2.1–2.4, respectively.

**2.1. Block Selection.** In this section, we try to find the best blocks for embedding the watermark. For this purpose, the original image is divided to  $a \times a$  nonoverlapping blocks and subsequently DCT is applied on each block. In the following of this paper, the value of  $a$  is considered as 8 to increase the method robustness versus compression because the standard JPEG is based on  $8 \times 8$  blocks. Then, the following properties of Human Visuals System (HVS) model that is suggested in [24, 28] is used for selecting blocks that are suitable for embedding watermark.

- (i) *Luminance Sensitivity* ( $L_k$ ). The brighter the background, the lower the visibility of the embedded watermark. It is estimated by following relation:

$$L_k = \frac{X_{DC,k}}{\bar{X}_{DC}}, \quad (1)$$

where  $X_{DC,k}$  is the DC coefficient of  $k$ th block and  $\bar{X}_{DC}$  is the mean value of DC coefficients of an original image.

- (ii) *Texture Sensitivity* ( $T_k$ ). The stronger the texture, the lower visibility of embedded watermark. It can be estimated by quantizing the DCT coefficients of a block ( $X_k$ ) using the JPEG quantization table ( $Q$ ). The latter results are then rounded to the nearest integers. The number of nonzero coefficients is then computed. This number presents the texture of that block:

$$T_k = \text{Number\_of\_non\_Zero} \left( \text{Round} \left( \frac{X_k}{Q} \right) \right), \quad (2)$$

where  $X_k$  are coefficients and Number\_of\_non\_Zero counts nonzero coefficients in  $k$ th block.

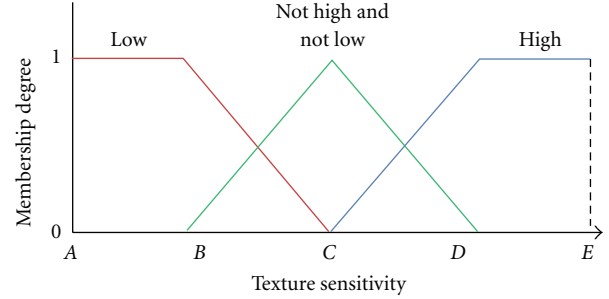


FIGURE 2: Dynamic membership function for texture sensitivity.

- (iii) *Location Sensitivity* ( $C_k$ ). The center quarter of an image is perceptually more important than other areas of the image. We estimate location of each block by computing the following ratio:

$$C_k = \frac{\text{center}(B_k)}{64}, \quad (3)$$

where  $\text{center}(B_k)$  is the number of pixels of the  $k$ th block lying in the center quarter (25%) of the image.

When these parameters are computed, they can be used to select blocks and determine weighting factor for embedding. In the proposed method, a Fuzzy Inference System (FIS) for calculating the relationship established between all properties of the HVS model is used because FIS provides simple mapping from a given set of inputs to another set of outputs without the complexity of mathematical modeling concepts.

Here, a DFIS has been used [28] and optimized in order to approximate the relationship established between three properties of the HVS model for both block selection and embedding process. We supposed the location sensitivity parameter is independent to images, therefore, in this model a static membership function is used for location sensitivity and only texture sensitivity and luminance sensitivity have dynamic membership functions. In the proposed DFIS, as it is shown in Figure 1, the inputs consist of texture sensitivity, luminance sensitivity, and location sensitivity parameters of each block and the outputs consist of corresponding suitability and weighting factors. The shape and support set values for inputs and outputs MFs (Membership functions) have been derived from experiments on various images.

The suitability parameter ( $\alpha$ ) depends to all three inputs but the weighting factor ( $\beta$ ) only depends on the texture sensitivity and luminance sensitivity. Let now explain for instance how the texture sensitivity membership function is

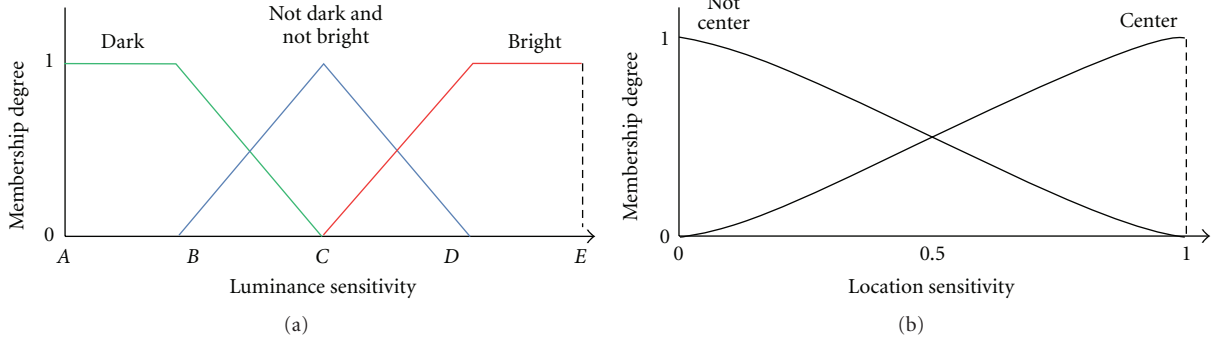


FIGURE 3: Membership functions for (a) luminance sensitivity and (b) location sensitivity.

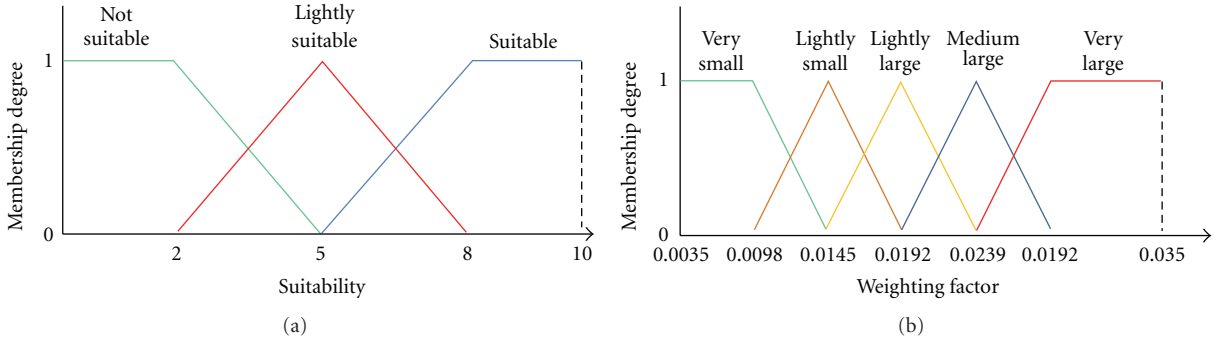


FIGURE 4: Membership Functions outputs of DFIS, (a) Suitability ( $\alpha$ ) and (b) Weighting Factor ( $\beta$ ).

computed. The structure of Texture sensitivity membership function has been shown in Figure 2. To compute the membership function parameters; first, we set  $A$  and  $E$  to take the minimum and maximum values of texture sensitivity  $T_k$  (4):

$$\begin{aligned} A &= \min(T_k), \\ E &= \max(T_k), \end{aligned} \quad (4)$$

where  $T_k$  is the Texture sensitivity of the  $k$ th block of the image. Then, in order to find point  $C$ , the average of the texture sensitivity of all  $8 \times 8$  blocks in the image is computed as shown in (5), where  $BN$  is the number of  $8 \times 8$  blocks in the image:

$$C = \sum_{k=1}^{BN} \left( \frac{T_k}{BN} \right). \quad (5)$$

Finally, points  $B$  and  $D$  are determined, in such a manner that these points never overlap or precede points  $A$  or  $E$ . The point  $B$  is equal to the median of texture values that are between  $A$  and  $C$  values as shown in (6), where  $T_k$  is the Texture of  $k$ th block and the point  $D$  is equal to the median of textures values that are between  $C$  and  $E$  values as shown in (7):

$$B = \text{median}\{T_k \mid A \leq T_k \leq C\}, \quad (6)$$

$$D = \text{median}\{T_k \mid C \leq T_k \leq E\}. \quad (7)$$

When points  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $E$  are determined, the slopes of all membership functions (MFs) are computed. The membership function of the other dynamic parameter (Luminance sensitivity) is created in the same way. Membership functions for Luminance sensitivity and Location sensitivity are shown in Figure 3. It is worth mentioning that the shape of location sensitivity membership functions is different from the others, because the experiments showed that this kind of MFs better fits to the used data than the others. So, the location sensitivity membership function ( $\mu_u$ ) is defined as the *Z-function*. The *Z-function* models this property using the following equation:

$$\mu_u = \begin{cases} 1, & u \leq p, \\ 1 - 2 \times \frac{(u-p)^2}{(q-p)}, & p < u \leq \frac{(p+q)}{2}, \\ 2 \times \frac{(u-q)^2}{(q-p)}, & \frac{(p+q)}{2} < u \leq q, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Figure 3(b) shows a plot of this function. In (8),  $p$  and  $q$  are two constant values that should be specified heuristically; for example, the best values that we found for *not center* curve were  $p = 0$  and  $q = 1$ . The same curves for all images have been used.

The membership functions for outputs of DFIS ( $\alpha$  and  $\beta$ ) are shown in Figure 4. After defuzzification we have crisp



FIGURE 5: (a) Original Lena, (b) Lena created using only DC coefficients of original image, (c) Lena created using estimate formulas (Gonzales method [3]) on image (b), and (d) Lena created using estimate formulas (Veeraswamy method [4]) on image (b).

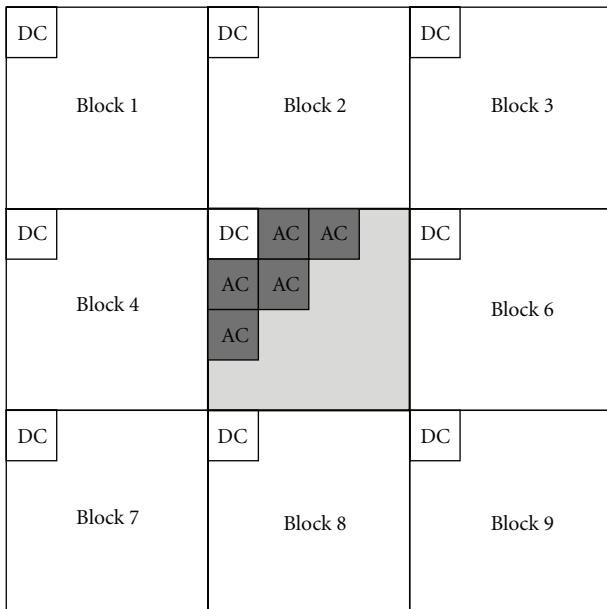


FIGURE 6: The position of five AC components in the central block that estimated using nine DC components of  $3 \times 3$  neighborhoods.

values for  $\alpha_k$  and  $\beta_k$  which determine the suitability and weighting factor of  $k$ th block of image. The blocks with highest  $\alpha_k$  values are selected for embedding process.

**2.2. Watermark Creation.** As we know, most of the signal energy of the block DCT is compacted in the DC component and the remaining energy is distributed diminishingly in the AC components in zigzag scan order [4]. On the other hand, the DC component is more robust than AC components versus different attacks. However, with having only DC coefficient in each block of an image, the overall look of that image is presented. For example, Figure 5(b) shows the Lena image that created using only DC component of each  $4 \times 4$  DCT block.

Since the DCT coefficients in one block for most images have a high correlation with the adjacent blocks; Gonzales

et al. in [3] described a technique which estimates a few low-frequency AC coefficients precisely. Moreover, only the DC values of  $3 \times 3$  neighborhoods of each central block are needed to estimate the AC coefficients of each central block as shown in Figure 6. The estimation relations for the first five AC coefficients of each  $4 \times 4$  DCT block are shown in Table 1 (first column) and Figure 5(c) shows the Lena image that created using these relations.

Gonzales et al. [3] did not consider variations in the image in AC coefficients estimation, but in [4] a new method was proposed that considered the variation in the image and accordingly AC coefficients are estimated with different equations. This method is better than Gonzales method in terms of reduced blocking artifacts and improved PSNR value. In this method, at first, the entropy of each block is calculated and then blocks with entropy values less than a threshold value are defined as smoother blocks and blocks with entropy values equal or greater than a threshold are considered as featured blocks. Based on entropy values, three cases are considered in estimation relations: (1) Smoother blocks, (2) Featured blocks and (3) Featured blocks surrounded by featured blocks. The estimation formulas based on Veeraswamy method to support  $4 \times 4$  DCT block for these three cases are shown in Table 1 (2th and 3th columns) and Table 2, correspondingly.

Based on this idea, only DC coefficients are needed to estimate the AC coefficients of each block [4]. Therefore, the estimating formulas (as shown in Tables 1 and 2 for  $4 \times 4$  DCT block) are employed to find these coefficients. Figure 5(d) shows a sample image that created by this method when the size of block is  $4 \times 4$ .

For watermark creation process, as shown in Figure 7, the original grayscale watermark image is divided into  $b \times b$  (e.g.,  $4 \times 4$ ) nonoverlapping blocks and subsequently performing the DCT on each block. Next, all AC coefficients are changed to zero.

In the proposed method, this created watermark image is inserted in the original image. In the extraction process, the estimating formulas (as shown in Tables 1 and 2 for  $4 \times 4$  DCT block) are employed to reconstruct the watermark.

TABLE 1: The formulas to estimate the first five AC coefficients of each  $4 \times 4$  DCT block (1).

Estimation formulas (Gonzales's method)	Estimation formulas for smoother blocks (Veeraswamy's method)	Estimation formulas for featured blocks (Veeraswamy's method)
$AC(0, 1) = 0.14125 \times (DC_4 - DC_6)$	$AC(0, 1) = 0.175 \times (DC_4 - DC_6)$	$AC(0, 1) = 0.231 \times (DC_4 - DC_6)$
$AC(1, 0) = 0.14125 \times (DC_2 - DC_8)$	$AC(1, 0) = 0.175 \times (DC_2 - DC_8)$	$AC(1, 0) = 0.231 \times (DC_2 - DC_8)$
$AC(0, 2) = 0.03485 \times (DC_4 + DC_6 - 2 \times DC_5)$	$AC(0, 2) = 0.083 \times (DC_4 + DC_6 - 2 \times DC_5)$	$AC(0, 2) = 0.118 \times (DC_4 + DC_6 - 2 \times DC_5)$
$AC(2, 0) = 0.03485 \times (DC_2 + DC_8 - 2 \times DC_5)$	$AC(2, 0) = 0.083 \times (DC_2 + DC_8 - 2 \times DC_5)$	$AC(2, 0) = 0.118 \times (DC_2 + DC_8 - 2 \times DC_5)$
$AC(1, 1) = 0.02026 \times (DC_1 + DC_9 - DC_3 - DC_7)$	$AC(1, 1) = 0.029 \times (DC_1 + DC_9 - DC_3 - DC_7)$	$AC(1, 1) = 0.15 \times (DC_1 + DC_9 - DC_3 - DC_7)$

TABLE 2: The formulas to estimate the first five AC coefficients of each  $4 \times 4$  DCT block (2).

Featured blocks surrounded by horizontal featured blocks (Veeraswamy's method)	Featured blocks surrounded by vertical featured blocks (Veeraswamy's method)	Featured blocks surrounded by horizontal and vertical featured blocks (Veeraswamy's method)
$AC(0, 1) = 0.231 \times (DC_1 - DC_3)$	$AC(0, 1) = 0.231 \times (DC_4 - DC_6)$	$AC(0, 1) = 0.231 \times (DC_1 - DC_3)$
$AC(1, 0) = 0.231 \times (DC_2 - DC_8)$	$AC(1, 0) = 0.231 \times (DC_1 - DC_7)$	$AC(1, 0) = 0.231 \times (DC_1 - DC_7)$
$AC(0, 2) = 0.15 \times (DC_1 - DC_3)$	$AC(0, 2) = 0.118 \times (DC_4 + DC_6 - 2 \times DC_5)$	$AC(0, 2) = 0.15 \times (DC_1 - DC_3)$
$AC(2, 0) = 0.118 \times (DC_2 + DC_8 - 2 \times DC_5)$	$AC(2, 0) = 0.15 \times (DC_1 - DC_7)$	$AC(2, 0) = 0.15 \times (DC_1 - DC_7)$
$AC(1, 1) = 0.15 \times (DC_1 + DC_9 - DC_3 - DC_7)$	$AC(1, 1) = 0.15 \times (DC_1 + DC_9 - DC_3 - DC_7)$	$AC(1, 1) = 0.15 \times (DC_1 + DC_9 - DC_3 - DC_7)$

2.3. *Watermark Insertion.* To describe the proposed method, we supposed that the original image ( $I$ ) and created watermark image ( $W'$ ) are grayscale images with size  $M \times N$  and  $L \times K$ , respectively.

In the watermark embedding process, the original image is transformed to frequency domain by DCT. Because the JPEG standard is based on  $8 \times 8$  block DCT, thus, block DCT with size of  $8 \times 8$  is commonly used in image watermarking process to make it robust versus JPEG compression [29]. Based on this idea, the original image is divided into  $8 \times 8$  nonoverlapping blocks and DCT is applied on each block. Next, the Dynamic Fuzzy Inference System (DFIS) is used to calculate the  $\alpha_k$  and  $\beta_k$  for each  $8 \times 8$  DCT block of original image. Then,  $(L/b \times K/b)$  number of blocks of original image with highest  $\alpha_k$  are selected for embedding watermark image, where  $b \times b$  is the size of DCT block of watermark. In the other side, the image that is created by described approach in Section 2.2 (used watermark) is divided into  $b \times b$  nonoverlapping blocks and then DCT is performed on each block. If  $b$  is smaller than  $a$  (in the proposed method, the value of  $a$  is 8), more robustness against attacks and more visual enhanced extracted watermark can be achieved but the quality of watermarked image is decreased. Thus,  $b$  provides a tradeoff between robustness after attacks and quality of watermarked image. Finally, the DC value of each  $b \times b$  DCT block of the created watermark is embedded in DC value of each selected  $8 \times 8$  DCT block of original image (based on (9)). Therefore, the watermarked image is created by modifying DC value of each  $8 \times 8$  DCT block of the original image. As shown in Figure 8, the following steps are used to insert the watermark in the original image.

*Algorithm 1* (The watermark embedding). We have the following.

**Input:** An original image  $I$ , watermark ( $W$ ).

**Output:** A watermarked image  $I_W$ .

*Step 1.* Divide the original image  $I$ , into  $8 \times 8$  nonoverlapping blocks and apply DCT on each block. Next, compute the HVS model properties as said in Section 2.1 and compute  $\alpha_k$  and  $\beta_k$  values of each block with Fuzzy Inference System as described in Section 2.1. Finally, sort blocks in descending order of  $\alpha_k$  value of each block.

*Step 2.* Create used watermark ( $W'$ ) from original watermark ( $W$ ) as described in Section 2.2.

*Step 3.* Select first  $(L/b \times K/b)$  blocks of sorted blocks which is computed in Step 1 for embedding process ( $L \times K$  is size of created watermark).

*Step 4.* Use (9) for invisible insertion of the created watermark (used watermark) into the DC coefficients of selected blocks of the original image:

$$X'_{DC,k} = X_{DC,k} + \varphi_k \times \beta_k \times W'_{DC,k}, \quad (9)$$

where  $X'_{k,DC}$  and  $X_{k,DC}$  are DC coefficients in  $k$ th block of watermarked image and original image, respectively and  $W'_{DC,k}$  is DC coefficient of  $k$ th block in created watermark. The  $\beta_k$  parameter is a weighting factor that



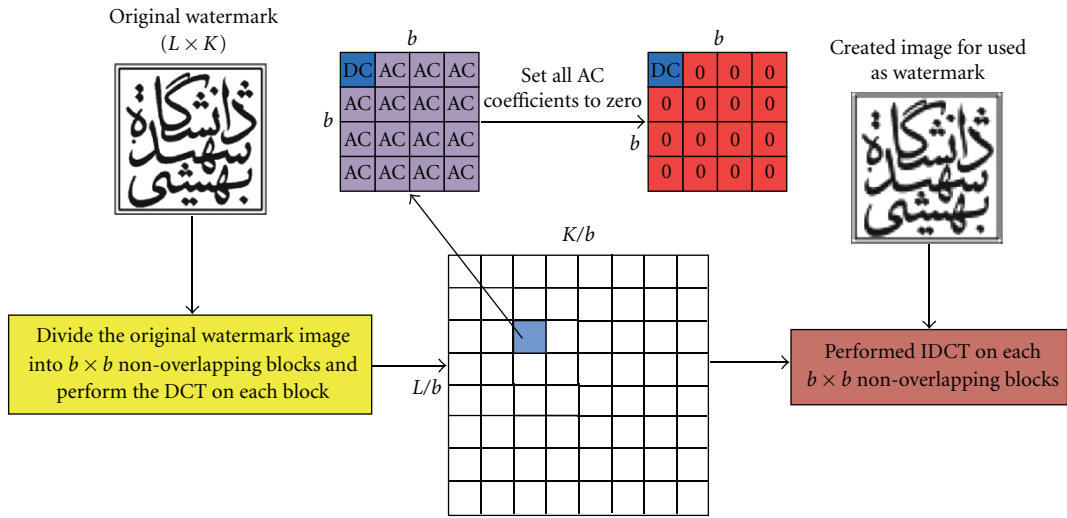


FIGURE 7: The watermark creation procedure.

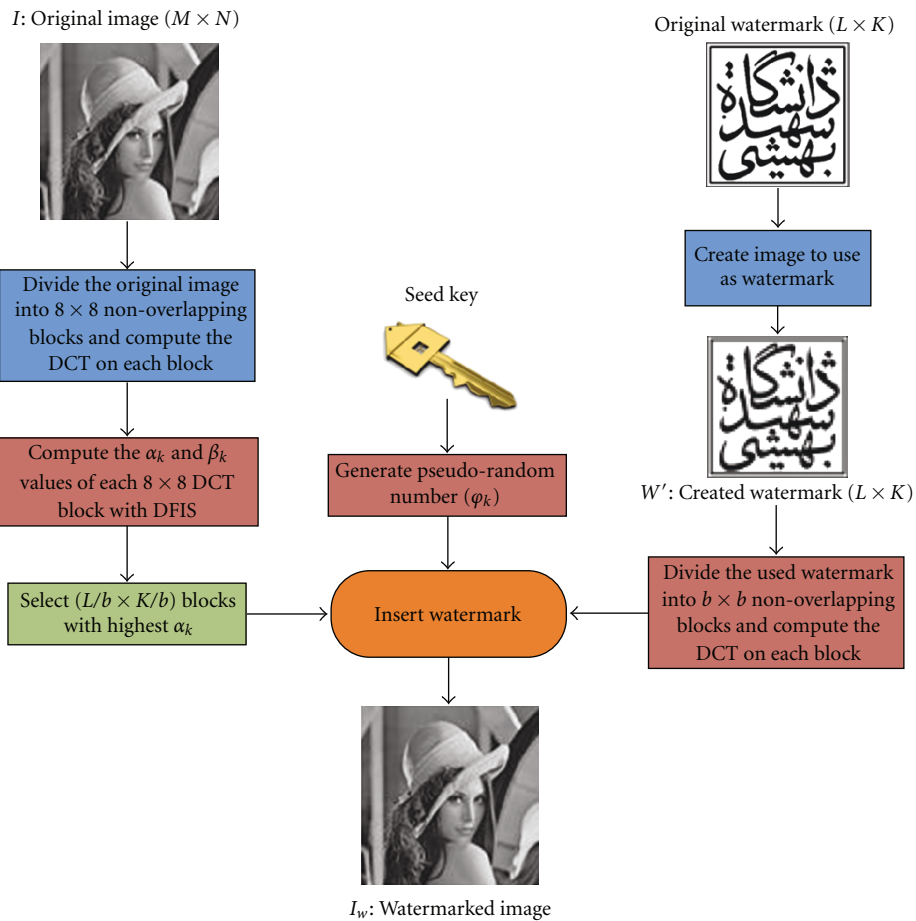


FIGURE 8: The proposed watermark embedding process.

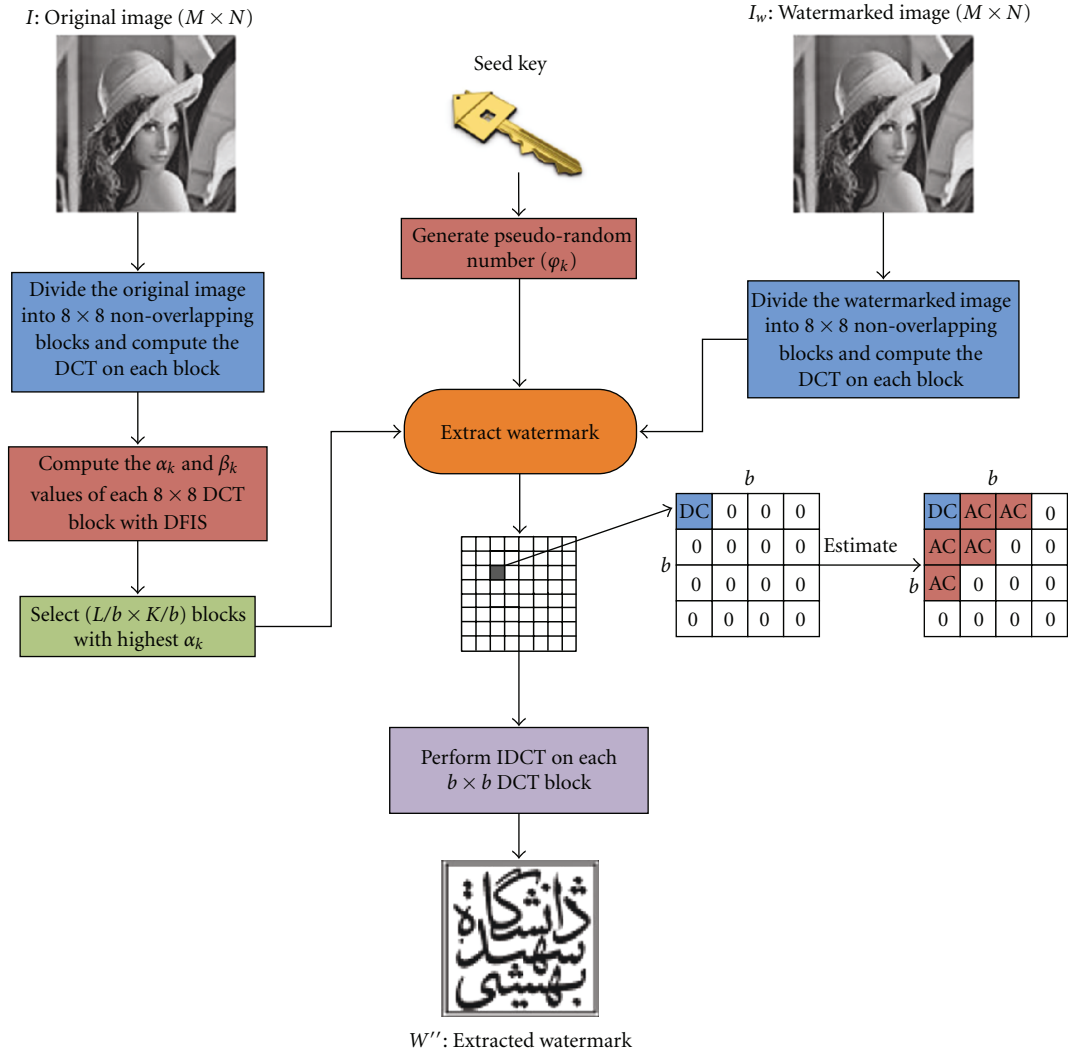


FIGURE 9: Proposed watermark extraction process.

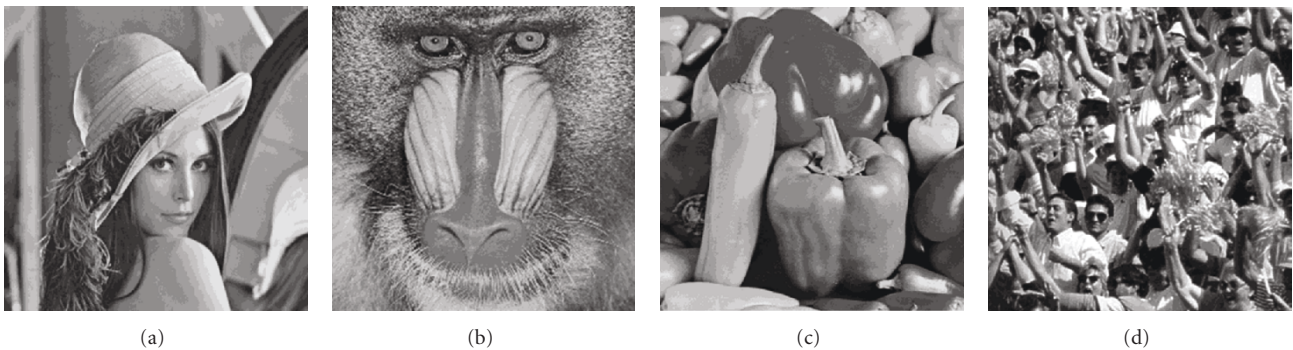


FIGURE 10: (a) Original Lena image, (b) Original Baboon image, (c) Original Peppers image and (d) Original Crowd image.

controls the tradeoff between invisibility, robustness, and detection fidelity of watermarked image which is computed by DFIS as described in Section 2.1. The  $\varphi_k$  parameter is a pseudorandom  $(1, -1)$  bit pattern that determines the addition or subtraction involved at each position which

can be any arbitrarily chosen pseudorandom sequence. This parameter is just used for security purpose.

*Step 5.* Use inverse DCT on each  $8 \times 8$  block to obtain watermarked image  $I_w$ .



FIGURE 11: (a) and (c) Original watermarks, (b) and (d) Used watermarks.



FIGURE 12: (a)–(d) Watermarked images after embedding  $128 \times 128$  watermark image as shown in Figure 11(b). (e)–(h) Extracted watermarks from (a)–(d), respectively. (i)–(l) Watermarked images after embedding  $128 \times 128$  watermark image as shown in Figure 11(d). (m)–(p) Extracted watermarks from (i)–(l), respectively.

2.4. Watermark Extraction. The watermark extraction process is the reverse of embedding process and requires the original image. As illustrated in Figure 9, at first, the watermarked image ( $I_w$ ) and the original one ( $I$ ) are divided into  $8 \times 8$  nonoverlapping blocks and the DCT is performed on each block of images. Next, as described in Section 2.1,  $\alpha_k$  and  $\beta_k$  values of each block in original image is computed

with DFIS and then the  $(L/b \times K/b)$  number of blocks with highest  $\alpha_k$  are selected ( $L/b \times K/b$  is number of  $b \times b$  blocks in watermark image and  $L \times K$  is size of it). Then, the DC coefficients of extracted watermark are computed as follows:

$$W''_{DC,k} = \frac{X''_{DC,k} - X_{DC,k}}{\varphi_k \times \beta_k}, \quad (10)$$





FIGURE 13: (a)–(d) Watermarked images after embedding  $256 \times 256$  watermark image as shown in Figure 11(b). (e)–(h) Extracted watermarks from (a)–(d), respectively. (i)–(l) Watermarked images after embedding  $256 \times 256$  watermark image as shown in Figure 11(d). (m)–(p) Extracted watermarks from (i)–(l), respectively.

where  $X''_{DC,k}$  and  $X_{DC,k}$  are DC coefficients of  $k$ th block in watermarked image and original image, respectively and  $W''_{DC,k}$  is DC coefficient of  $k$ th block in extracted watermark. The  $\beta_k$  parameter is a weighting factor which is computed in Step 1 and the  $\varphi_k$  parameter is a pseudorandom (1, -1) bit pattern that generated with arbitrary seed and used in insertion process.

Finally, the  $W''_{DC,k}$  values and estimation formulas as described in Section 2.2 are used to create the  $b \times b$  DCT blocks of watermark then by performing Block-wise inverse DCT, watermark in spatial domain is created. The following steps are used for watermark Extraction.

*Algorithm 2* (The watermark extraction). We have the following.

**Input:** An original image ( $I$ ) and watermarked image ( $I_W$ ).

**Output:** An extracted watermark ( $W''$ ).

*Step 1.* Divide the original image into  $8 \times 8$  nonoverlapping blocks and compute the DCT on each block. Then compute the HVS model properties as said in Section 2.1 and compute  $\alpha_k$  and  $\beta_k$  values of each block with fuzzy approach (DFIS). Finally, sort blocks in descending order of  $\alpha_k$  value of each block.





FIGURE 14: (a)–(d) The watermarked Lena, Baboon, Peppers and Crowd images with  $256 \times 256$  watermark image as shown in Figure 11(b) after JPEG compression with quality factor 40%, 30%, 20% and 10%, respectively. (e)–(h) The extracted watermarks ( $256 \times 256$ ) from (a)–(d), respectively. (i)–(l) The watermarked Lena, Baboon, Peppers and Crowd images with  $256 \times 256$  watermark image as shown in Figure 11(d) after JPEG compression with quality factor 40%, 30%, 20% and 10%, respectively. (m)–(p) The extracted watermarks ( $256 \times 256$ ) from (i)–(l), respectively.

*Step 2.* Select first  $(L/b \times K/b)$  blocks of sorted blocks which is computed in Step 1 for extracting process.  $L/b \times K/b$  is number of  $b \times b$  blocks in watermark and  $L \times K$  is size of watermark.

*Step 3.* Divide the watermarked image  $I_W$  into  $8 \times 8$  nonoverlapping blocks and compute the DCT on each block.

*Step 4.* Extract the watermark from selected blocks use (10).

*Step 5.* Estimate AC coefficients of each block in extracted watermark from Step 4, then use  $b \times b$  Block-wise inverse DCT to create extracted watermark in spatial domain ( $W''$ ). If the input watermark image is present in the extracted image, then the ownership is approved.

### 3. Structural Comparison of Proposed Method with Related Ones

The employed techniques in proposed method make it more robust and its results with more quality. In this section, the differences and excellences of proposed method with two related methods [24, 27] are introduced in four conventional different steps of watermarking methods: (1) selecting embedding area procedure, (2) watermark creation procedure, (3) inserting procedure, (4) extracting procedure. Also, the motivation of proposed method is implied in subsections.

*3.1. Selecting Embedding Area Procedure.* Mohanty's method [24], at first, finds the most perceptually important subimage



FIGURE 15: (a)–(d) The watermarked Lena, Baboon, Peppers and Crowd images with  $128 \times 128$  watermark image as shown in Figure 11(b) after JPEG compression with quality factor 40%, 30%, 20% and 10%, respectively. (e)–(h) The extracted watermarks ( $128 \times 128$ ) from (a)–(d), respectively. (i)–(l) The watermarked Lena, Baboon, Peppers and Crowd image with  $128 \times 128$  watermark images as shown in Figure 11(d) after JPEG compression with quality factor 40%, 30%, 20% and 10%, respectively. (m)–(p) The extracted watermarks ( $128 \times 128$ ) from (i)–(l), respectively.

of original image, where the size of subimage is equal to size of watermark ( $L \times K$ ) to embed the watermark in it. To find this subimage, the properties of Human Visual System (HVS) such as Luminance, Edginess, Contrast, Location and Texture are calculated for each  $L \times K$  subimage of original image and the high score one is selected as most perceptually important region of original image and watermark is embedded in it. As result, this method is not robust to geometrical attacks such as Tampering, Data block removal and Cropping; because the watermark is embedded in consecutive blocks (subimage) of original image. For example if this region of watermarked image is cropped or tampered, the whole watermark is removed and the extraction procedure cannot find any watermark in the test image (see Section 4.3). But, in the proposed method, blocks of the watermark are not embedded in consecutive blocks of original image and embedded in nonconsecutive blocks of original image. As result, the proposed method is more robust versus many

geometrical attacks such as Tampering, Data block removal, and Cropping (see Sections 4.2 and 4.3).

In Asatryan's method [27] that inserts the watermark in spatial domain, all pixels of original image are used to embed the watermark. Therefore, the quality of watermarked image in this method is degraded and artifact is produced in watermarked image (see Section 4.3).

**3.2. Watermark Creation Procedure.** Mohanty's method create synthetic image by using  $8 \times 8$  DCT coefficients of selected subimage of original image and Gaussian, Laplacian distributions for DC, AC coefficients, respectively. Then, the original watermark is embedded in the created synthetic image using any DCT-based visible watermarking algorithm to create used watermark.

In the Asatryan's method, the used watermark is created by compressing the original watermark that the rate of compression is defined by user.



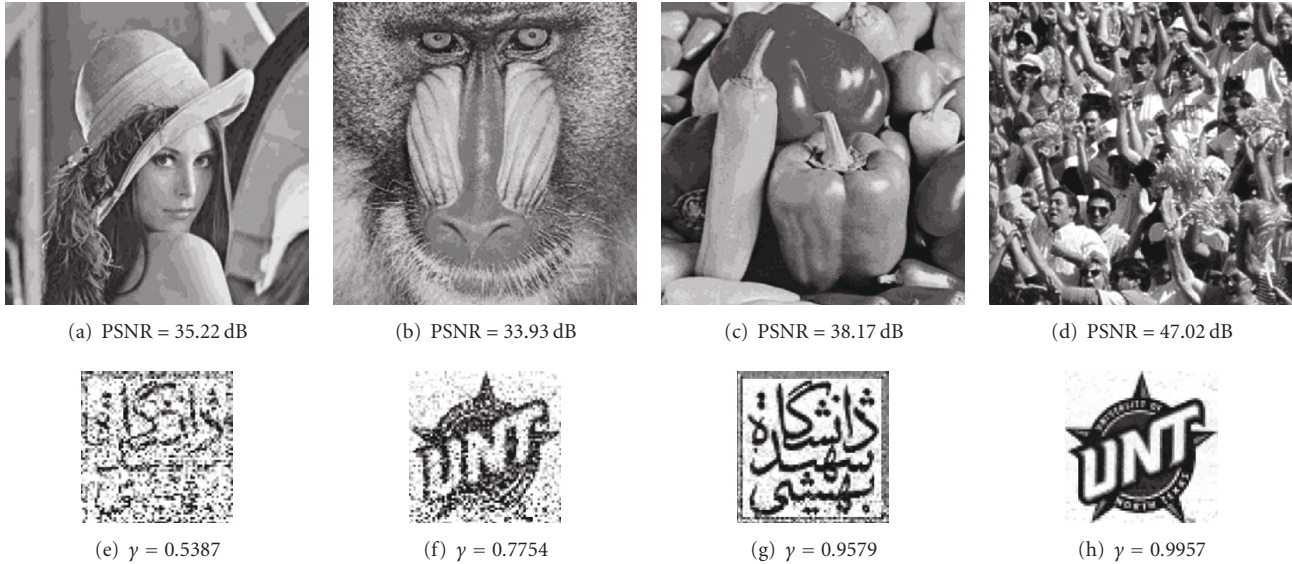


FIGURE 16: (a)–(d) The watermarked Lena, Baboon, Peppers and Crowd images with  $256 \times 256$  watermark image after wavelet compression with quality factor 0.4 bpp, 0.8 bpp, 1.5 bpp and 3.5 bpp, respectively. (e)–(h) The extracted watermarks ( $256 \times 256$ ) from (a)–(d), respectively.

In the proposed method the used watermark is created by dividing the original watermark into  $b \times b$  DCT blocks and changing the AC coefficients of each block to zero. The  $b$  parameter provides a tradeoff between quality of watermarked image and extracted watermark. The proposed creation watermark procedure is acceptable because the watermark image that creates by only DC coefficients of each  $b \times b$  (where  $b < 8$ ; e.g.,  $b = 4$ ) DCT block of original watermark is perceptually similar as original one. Also, the AC coefficients estimating formulas that propose in [4] can be used to increase the quality of created watermark.

**3.3. Inserting Procedure.** In the Mohanty’s method the used watermark is embedded into the original image by fusing the DCT coefficients of used watermark blocks with the corresponding blocks of the selected subimage. In the other hand, the DCT coefficients of each  $8 \times 8$  DCT block of used watermark is embedded in corresponding  $8 \times 8$  DCT block of selected subimage. As result, the robustness of mohanty’s method decreases because the AC coefficients of DCT block is not robust to many attacks such as Low Pass Filtering, Compression, Median Filtering. Therefore, the many of embedded AC coefficients of used watermark are degraded after such attacks. To solve this drawback, in the proposed method, the coefficients of  $b \times b$  (where  $b < 8$ ) DCT blocks of used watermark are embedded only in DC coefficients of each  $8 \times 8$  DCT block of original image. As result, the robustness of proposed method is higher than mohanty’s method, because the DC coefficients of DCT block is robust than AC coefficients of one.

The Asatryan’s method works in spatial domain to embed the watermark in original image. In this method, the values of  $32 \times 32$  block DCT coefficients of compressed watermark are mapped to the interval  $[0,255]$  by fixed linear transform

and the mapped values of DCT coefficients are embedded in pixel values of each block of original image. As result, because the embedding is done in special domain, the robustness of this method is decreased and the quality of watermarked image is low (see Section 4.3). Also, mapping the DCT coefficients to the interval  $[0,255]$  may be caused distortion in the extracted watermark.

The weighted factor ( $\beta$ ) is used in all three methods. The value of this parameter is 0.02 for DC and 0.1 for AC coefficients in Mohanty’s method and 0.07 for all pixels in Asatryan’s method. But, in the proposed method, the value of this parameter for each DCT block is based on Texture and Luminance of this block. It is based on idea that modification inside a highly textured block is unnoticeable to the human eye and the brighter the background is the lower the visibility of the embedded watermark. Therefore, the proposed method produces a watermarked image with higher quality than two related methods.

**3.4. Extracting Procedure.** The Mohanty’s method use a reverse embedding procedure to extract the DCT coefficients of each  $8 \times 8$  DCT block of watermark and applied IDCT to create watermark in spatial domain. But in proposed method a reverse embedding procedure is performed to extract the only DC coefficients of each  $b \times b$  DCT block of watermark. Then the estimation formulas are used to evaluate the AC coefficients of each  $b \times b$  DCT block (e.g., first five AC coefficients when  $b = 4$ ) of watermark and applied IDCT to create watermark in spatial domain.

The Asatryan’s method use a reverse embedding procedure (in spatial domain) to extract the mapped DCT coefficients of watermark. Then the reverse of linear transform that used in embedding process is used to create the DCT coefficients of watermark. Finally, IDCT is applied to create



FIGURE 17: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after adding Gaussian noise with mean = 0 and variance = 0.002, (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after adding Gaussian noise. (g) and (h) The extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after adding Gaussian noise, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after adding Gaussian noise. (o) and (p) The extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

the watermark in spatial domain. The steps of Mohanty's method, Asatryan' method and proposed watermarking method are summarized in Table 3.

#### 4. Experimental Results

The proposed algorithm has been tested on different images and a big set of grayscale watermark images but only results for four popular images and two logos with different sizes are presented here. The selected logos are *Texas University*

and *ShahidBeheshti University* ones. We have chosen *Lena*, *Baboon*, *Peppers* and *Crowd* grayscale images with size  $512 \times 512$  as shown in Figure 10 to embed watermarks in them and the watermarks are grayscale watermark logos with size  $128 \times 128$  and  $256 \times 256$  as shown in Figure 11. Also, based on experiments on different watermark images (with size  $128 \times 128$  and  $256 \times 256$ ), the value of  $b$  was selected equal to 4. The program development tool was MATLAB and the computation platform was a personal computer with 1.66 GHZ of CPU and 2 GB of RAM.



TABLE 3: Comparing the structures of D. Asatryan and N. Asatryan [27], Mahanty and Bhargava [24], and proposed watermarking methods.

		Watermarking methods	
Block size of original image	Mohanty's method [24]	Asatryan's method [27]	Proposed method
	$8 \times 8$	$M/K \times N/L$	$8 \times 8$
Selection embedding area procedure	(1) Find a most perceptually significant set of blocks constituting a subimage (equal to size of watermark) with respect to human perception such as Texture, Location, Contrast, Luminance, and Edginess in original image.	(1) All blocks of original image are used to embed a watermark	(1) Calculate the Texture, Luminance, and Location of each $8 \times 8$ DCT block in original image. (2) Use proposed Fuzzy Interface System to calculate the suitability factor of each block. (3) Select $(L/b \times K/b)$ blocks with higher suitability factor to embed a watermark.
Block size of watermark	$8 \times 8$	$32 \times 32$	$b \times b$
Watermark creation procedure	(2) Create synthetic image by using $8 \times 8$ DCT coefficients of most perceptually important subimage of original image and Gaussian/Laplacian distribution for DC, AC coefficients, respectively. (3) Embed the watermark in the synthetic image using any DCT-based visible watermarking algorithm.	(2) Compression was performed on watermark image until the number of chosen DCT coefficients of each $32 \times 32$ DCT block was significantly smaller than the number of pixels of the original watermark. (3) The values of DCT coefficients are mapped to the interval [0,255] by fixed linear transform.	(4) Change the AC coefficients of each $b \times b$ DCT block to zero and apply IDCT to create used watermark.
Inserting procedure	(4) The used watermark is now invisibly embedded into the original image by fusing the compound watermark blocks with the corresponding blocks of the selected perceptually important subimage of the original.	(4) Embed each mapped DCT coefficient of watermark in each pixel of block of original image.	(5) Embed DC coefficient of each $b \times b$ DCT block of used watermark in DC coefficient of selected $8 \times 8$ DCT block of original image.
Value of $\beta$ coefficient (weighing factor)	$\beta_{i,j,k} = \begin{cases} 0.02 & \text{for DC coefficients} \\ 0.1 & \text{for AC coefficients} \end{cases}$	$\beta = 0.07$ for all pixels	$\beta_k$ is different for each $8 \times 8$ DCT block of original image and is computed by proposed Fuzzy Interface System based on Texture and Luminance of selected block.
Extracting procedure	(1) Select subimage of original image where the watermark was embedded in it. (2) Use the reverse embedding procedure to extract the DCT coefficients of watermark. (3) Apply IDCT on each extracted $8 \times 8$ DCT block to create watermark in spatial domain.	(1) Use the reverse embedding procedure to extract the mapped DCT coefficients of watermark. (2) The reverse of linear transform where used in embedding process is utilized to create the DCT coefficients of watermark. (3) Apply IDCT on each extracted DCT block to create watermark in spatial domain.	(1) Select blocks of original image where the watermark was embedded in them. (2) Use the reverse embedding procedure to extract the DC coefficients of each $b \times b$ DCT block of used watermark. (3) The extracted DC coefficients are used to estimate the AC coefficients of each $b \times b$ DCT block of watermark. (4) Apply IDCT on each estimated $b \times b$ DCT block to create watermark in spatial domain.

The experiments confirmed the effectiveness of the proposed algorithm in producing visually pleasing watermarked images and in addition the extracted watermark was visually recognizable and similar to both inserted watermark and original watermark. Our scheme requires one key as seed of random number generator to be stored for extraction phase, so this method has no storage overhead. After the watermark is embedded into the original image, the PSNR (Peak Signal to Noise Ratio) is used to evaluate the quality

of the watermarked image. The MSE and PSNR values in decibels (dB) are defined as follows:

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right), \quad (11)$$

$$\text{MSE} = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (X_{ij} - X'_{ij})^2,$$



FIGURE 18: (a) and (b) The Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after performing Gaussian lowpass filter with window size  $5 \times 5$ , (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after performing Gaussian lowpass filter, (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after performing Gaussian lowpass filter, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after performing Gaussian lowpass filter, (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

where  $X_{ij}$  represents the  $(i, j)$  pixel value of original image and  $X'_{ij}$  represents the  $(i, j)$  pixel value of watermarked image. The other metric used to test the quality of the retrieved watermark image is Normalized Cross Correlation (NCC). It is defined as follows:

$$\gamma = \frac{\sum_{i,j} (w''_{i,j} - \bar{w}'') (w_{i,j} - \bar{w})}{\sqrt{\sum_{i,j} (w''_{i,j} - \bar{w}'')^2 \sum_{i,j} (w_{i,j} - \bar{w})^2}}, \quad (12)$$

where  $w''$  and  $w$  are extracted watermark and inserted watermark images, respectively, and  $\bar{w}''$  and  $\bar{w}$  are their pixels mean values, respectively. The subscripts  $i, j$  of  $w''$  or  $w$  denote the index of an individual pixel of the corresponding image. The summations are over all the image pixels.

The other part of experiments involved testing the algorithm against many common attacks on watermarked image and fortunately the extracted watermark in almost cases was detectable and acceptable due to the original and inserted watermark. In these experiments, we used





FIGURE 19: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after performing Average filter with window size  $5 \times 5$ , (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after performing Average filter, (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after performing Average filter, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after performing Average filter, (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

*StirMarkMATLAB* software that contains approximately 90 different types of image manipulations. But, in the following subsections, we will present only the experimental results for test images, and nongeometric and geometric attacks such as Compression, Noise addition, Filtering, Cropping, Changing Aspect Ratio, Tampering and Scaling on the watermarked images to evaluate the robustness of the proposed scheme.

4.1. Quality of Watermarked Image and Extracted Watermark before Attack. Four selected images that used in embedding

processare shown in Figure 10. Also, we used two watermarks in Figure 11 at two sizes ( $128 \times 128$  and  $256 \times 256$ ) to be embedded in these original images. The watermarked images and the extracted watermarks with corresponding PSNR with different size of watermarks ( $128 \times 128$  and  $256 \times 256$ ) are shown in Figures 12 and 13, respectively.

It is obvious that the PSNR value of the watermarked image had a higher value in comparison with other existing watermarking algorithms. The average PSNR value for the watermarked images was approximately 52 dB, where the size of watermark images is  $128 \times 128$ . Also, the average



FIGURE 20: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after Blurring with radius 3, (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after Blurring, (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after Blurring, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after Blurring, (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

PSNR value for the watermarked images was approximately 49 dB, where the size of watermark images is  $256 \times 256$ . So, the watermark embedding process produced high-quality watermarked images.

**4.2. Quality of Watermarked Image and Extracted Watermark versus Various Attacks.** In the following experiment, we used several image manipulations, including Compression, Noise addition, Filtering, Cropping, Changing aspect ratio, Tampering, Copy attack, Scaling and Composite attacks on the watermarked images to evaluate the robustness of the proposed scheme.

#### 4.2.1. Compression

**JPEG Compression.** Using image compression before storing and transmitting images is very common. JPEG from *Joint Photographic Experts Group* has been funded its way through digital imaging and is very popular image compression tool for still images. So we evaluated the robustness of the proposed scheme by compressing the watermarked images with different JPEG quality factors. Figures 14(a)–14(d), 14(i)–14(l) show the watermarked images with watermark size  $256 \times 256$  after JPEG compression with quality factor 40%, 30%, 20% and 10% for Lena, Baboon, Peppers and Crowd images, respectively. Figures 14(e)–14(h), 14(m)–14(p) show





FIGURE 21: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after Sharpening, (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after Sharpening, (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after Sharpening, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after Sharpening, (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

the extracted watermark from Figures 14(a)–14(d), 14(i)–14(l), respectively. Also, Figures 15(a)–15(d), 15(i)–15(l) show the watermarked images with watermark size  $128 \times 128$  after JPEG compression with quality factor 40%, 30%, 20% and 10% for Lena, Baboon, Peppers and Crowd images, respectively and Figures 15(e)–15(h), 15(m)–15(p) show the extracted watermark from Figures 15(a)–15(d), 15(i)–15(l), respectively. The results show that the proposed scheme is robust against JPEG image compression and the extracted watermarks are visually similar to inserted watermark under different quality factors of JPEG compression.

*Wavelet Compression (JPEG2000).* We evaluated the robustness of proposed method against another version of compression that is wavelet compression. Figures 16(a)–16(d) show the results of applying wavelet compression on Lena, Baboon, Peppers and crowd images with compression ratio 0.4 bpp, 0.8 bpp, 1.5 bpp and 3.5 bpp, respectively. The extracted watermarks that shown in Figures 16(e)–16(h) are still visually detectable after this attack.

*4.2.2. Noise Addition.* The robustness of proposed method has been evaluated by adding Gaussian noise with mean =



FIGURE 22: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after applying Median filter with window size  $5 \times 5$ , (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after applying Median filter, (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after applying Median filter, (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after applying Median filter, (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

0 and variance = 0.002 on the watermarked images. Figures 17(a)–17(d), 17(i)–17(l) show the results of adding Gaussian noise. The extracted watermarks are still visually detectable after this attack (as shown in Figures 17(e)–17(h), 17(m)–17(p)). It indicates that the proposed scheme is also robust to noise attack.

**4.2.3. Filtering.** The robustness of watermarking scheme has been also tested by performing various filters such as sharpening, Gaussian lowpass filter, averaging, median, and blurring on the watermarked images. Figures 18(a)–18(d),

18(i)–18(l) show the resultant images after performing Gaussian lowpass filter with window size  $5 \times 5$ . Figures 18(e)–18(h), 18(m)–18(p) show the extracted watermarks and corresponding  $\gamma$  values. The extracted watermarks are still visually detectable after this attack. It indicates that the proposed scheme is also robust to Gaussian lowpass filter attack.

Figures 19(a)–19(d), 19(i)–19(l) show the resultant images by averaging filter with window size  $5 \times 5$ . Figures 19(e)–19(h), 19(m)–19(p) show the extracted watermarks and their  $\gamma$  values. The extracted watermarks are still visually detectable after averaging filter attack.





FIGURE 23: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after Scaling (1/2), (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after Scaling (1/2), (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after Scaling (1/2), (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after Scaling (1/2), (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

Figures 20(a)–20(d), 20(i)–20(l) show the resultant images after blurring with radius 3. Figures 20(e)–20(h), 20(m)–20(p) show the extracted watermarks and their  $\gamma$  values. The extracted watermarks are still visually detectable after this attack. It indicates that the proposed scheme is also robust to blurring attack.

Figures 21(a)–21(d), 21(i)–21(l) show the resultant images by sharpening. Figures 21(e)–21(h), 21(m)–21(p) show the extracted watermarks. Also, Figures 22(a)–22(d), 22(i)–22(l) show the resultant images after median filtering with window size  $5 \times 5$ . Figures 22(e)–22(h), 22(m)–22(p)

show the extracted watermarks and their  $\gamma$  values. The test results show that the watermark image can also detectable after the filter attacks. (it is worth mentioning that because we zoomed out images in the paper the effects of some filters are not visible in these sizes).

4.2.4. Geometric Attacks. In the following experiments, different geometric attacks such as scaling, cropping, tampering and changing aspect ratio are performed on the watermarked images to test the robustness of proposed method.





FIGURE 24: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) after Scaling (1/4), (e) and (f) the extracted watermarks with size  $256 \times 256$  from (a) and (b), respectively. (c) and (d) Crowd and Baboon watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(b) after Scaling (1/4), (g) and (h) the extracted watermarks with size  $128 \times 128$  from (c) and (d), respectively. (i) and (j) Lena and Peppers watermarked images with  $128 \times 128$  watermark image as shown in Figure 11(d) after Scaling (1/4), (m) and (n) the extracted watermarks with size  $128 \times 128$  from (i) and (j), respectively. (k) and (l) Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d) after Scaling (1/4), (o) and (p) the extracted watermarks with size  $256 \times 256$  from (k) and (l), respectively.

*Scaling.* In this experimental the watermarked images are reduced to 1/2 and 1/4 of its original size. In order to detect the watermark, the reduced images are recovered to its original dimension, respectively. Figures 23(a)–23(d), 23(i)–23(l) show the watermarked image after reducing to 1/2 and recovering to original dimension. Figures 23(e)–23(h), 23(m)–23(p) show the extracted watermark from Figures 23(a)–23(d), 23(i)–23(l), respectively.

Figures 24(a)–24(d), 24(i)–24(l) show the watermarked image after reducing to 1/4 and recovering to original dimension. Figures 24(e)–24(h), 24(m)–24(p) show the

extracted watermark from Figures 24(a)–24(d), 24(i)–24(l) and corresponding  $\gamma$  values, respectively. The test results show that the watermark image can also detectable after the scaling attacks.

*Cropping.* In this experimental the watermarked images are cropped. Figures 25(a)–25(d) show the cropped version of Lena, Baboon, Peppers and Crowd watermarked image respectively. Figures 25(e)–25(h) show the extracted watermark from these figures. As shown from these figures, the extracted watermarks are visually detectable yet.

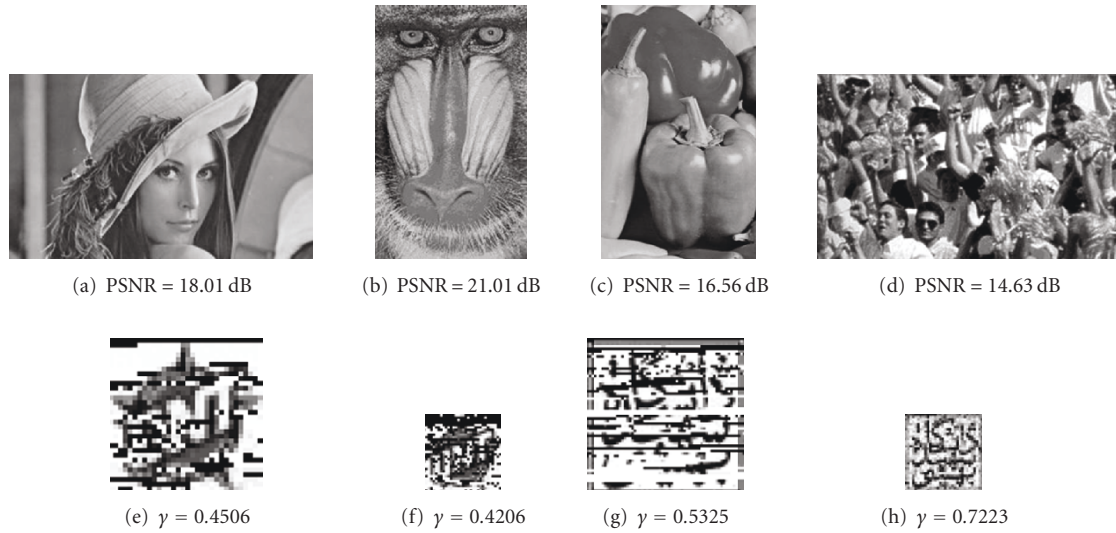


FIGURE 25: (a)–(d) the watermarked Lena, Baboon, Peppers and Crowd images after cropping, respectively. (e)–(h) the extracted watermarks from (a)–(d), respectively.

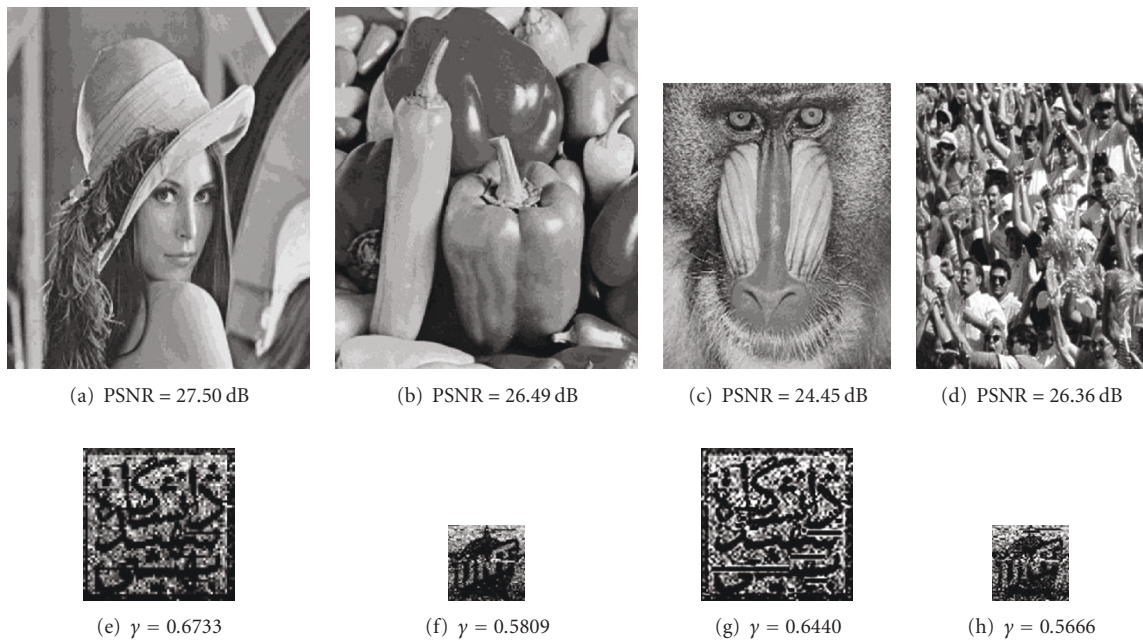


FIGURE 26: (a) and (b) The watermarked Lena and Peppers images after changing aspect ratio attack ( $X = 1, Y = 1.2$ ), respectively. (c) and (d) the watermarked Baboon and Crowd images after changing aspect ratio attack ( $X = 0.8, Y = 1$ ), respectively. (e)–(h) The extracted watermarks from (a)–(d), respectively.

*Changing Aspect Ratio.* In this experiment, the robustness of proposed method was tested by changing aspect ratio of watermarked image. Figures 26(a) and 26(b) show the Lena and Peppers images after changing aspect ratio ( $X = 1, Y = 1.2$ ) of these images and Figures 26(c) and 26(d) show the Baboon and Crowd images after changing aspect ratio ( $X = 0.8, Y = 1$ ) of these images. To extract the watermark, the images were rescaled to original size ( $512 \times 512$ ) and the extracted watermarks from these figures are shown in Figures 26(e)–26(h).

*Tampering and Data Blocks Removal.* We tested the robustness of proposed method by tampering the watermarked images. Figures 27(a)–27(d) show the results of tampering Lena, Baboon, Peppers and Crowd images, respectively. As shown in Figures 27(e)–27(h), the extracted watermarks are still visually detectable after this attack and it indicates that the proposed scheme is also robust to such attacks. Also, Figures 27(i)–27(l) show the results of data blocks removing of Lena, Baboon, Peppers and Crowd images, respectively and Figures 27(m)–27(p) show the extracted watermark





FIGURE 27: (a)–(d) The watermarked Lena, Baboon, Peppers and Crowd images after tampering, respectively. (e)–(h) The extracted watermarks from (a)–(d), respectively. (i)–(l) The watermarked Lena, Baboon, Peppers and Crowd images after data blocks removal, respectively. (m)–(p) The extracted watermarks from (i)–(l), respectively.

from Figures 27(i)–27(l), respectively. As result, the extracted watermark after such attacks are still visually detectable and the proposed method is robust to tampering and data blocks removal.

**Copy Attack.** The copy attack has been used to create the false positive problem and operated as follow: (1) a watermark is first predicted from watermarked image, (2) the predicted watermark into a target image to create counterfeit watermarked image, (3) from the counterfeit image, a watermark can be detected that wrongly claims rightful ownership.

In this experiment, the robustness of proposed watermarking method was tested by applying copy attack on

watermarked image. Figures 28(a) and 28(b) show the Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b) and Figures 28(c) and 28(d) show the Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d). Figures 28(e)–28(h) show the counterfeit watermarked image with Figures 28(a)–28(d), respectively. The extracting watermarks from Figures 28(e)–28(h) are shown in Figures 28(i)–28(l), respectively. Therefore, the proposed method is robust against copy attack.

**4.2.5. Composite Attacks.** The purpose of this experiment is to check whether this kind of combination attack is able to remove the watermark of the proposed method.





FIGURE 28: (a) and (b) Lena and Peppers watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(b), (e) and (f) the counterfeit watermarked image with (a) and (b), respectively. (c) and (d) The Crowd and Baboon watermarked images with  $256 \times 256$  watermark image as shown in Figure 11(d), (g) and (h) the counterfeit watermarked image with (c) and (d), respectively. (i)–(l) the extracted watermark from (e)–(h), respectively.

Figures 29(a)–29(h) show the watermarked images after different composite attacks and Figures 29(i)–29(p) show the extracted watermarks from Figures 29(a)–29(h), respectively.

Therefore, the experimental results presented on the quality and recognize ability demonstrates the performance of our method under various attacks.

**4.3. Comparison with Other Related Methods.** In this subsection, the results of proposed method are compared with two related ones which have been presented by Mahanty and Bhargava [24] and D. Asatryan and N. Asatryan [27]. The comparison is based on four metrics: (1) average execution time for watermark insertion (2) PSNR value of watermarked image, (3) PSNR or correlation value ( $\gamma$ ) value of extracted watermark and (4) error rate of detecting watermark.

These three methods were implemented on a personal computer with 1.66 GHZ of CPU and 2 GB of RAM and the average execution time of proposed method for watermark insertion was approximately 2 sec for an image with size 512

$\times 512$  pixels and watermark image with size  $128 \times 128$  pixels. The execution time for Mohanty method was 4 sec that is approximately 50% higher in time than proposed algorithm and 1 sec for Asatryan method that is approximately 50% lower than proposed algorithm.

Based on experiments, in the proposed method, the average minimum value of  $\gamma$  was 0.4 when the extracted watermark was visually detectable. This value for Mohanty method and Asatryan method were 0.65 and 0.3, respectively.

To have a complete comparison between proposed method and related ones, we embedded the 50 different watermark images in three sizes ( $64 \times 64$ ,  $128 \times 128$  and  $256 \times 256$ ) in 50 selected images in two sizes ( $256 \times 256$  and  $512 \times 512$ ) and obtained  $50 \times 50 = 2500$  watermarked images. Then we used *StirMark* and did different attacks to the watermarked images including Blurring, Sharpening, Scaling, adding Gaussian noise, Tampering, data block removal and Cropping. In addition, JPEG compression with different quality factors was applied to the watermarked images. Then, we conducted the watermark detection procedure on every attacked watermarked image. Table 4 shows the



FIGURE 29: (a)–(h) The watermarked image after different composite attacks, (i)–(k) The extracted watermark from (a)–(h), respectively.

PSNR of watermarked images and extracted watermarks. As it is shown in this table, the proposed method outperforms than two related methods in term of PSNR of watermarked images and extracted watermark after different attacks.

Finally, as Table 5 shows, the comparison results have demonstrated that our method is capable of detecting watermarks at lower error rates than two related methods and can more effectively stay robust under image processing attacks. Also, Table 6 shows PSNR value of watermarked image by different methods. The best value in each row of these tables has been bolded.

The quality of extracted watermark by proposed method and two related ones versus different attacks are summarized

in Tables 7, 8, 9 and 10 for Lena and Baboon with watermark size  $256 \times 256$ , Peppers and Crowd with watermark size  $128 \times 128$ , respectively. The 2th column of each of these tables represents the attack type and the symbols “AF”, “B”, “GN”, “MF”, “S”, “GL”, “SH”, “C”, “CAR”, “WF” and “JP” denote average filter, blurring, Gaussian noise, median filter, scaling, Gaussian lowpass filter, sharpening, cropping, change aspect ratio, wiener filter and JPEG compression, respectively. The number following each symbol is the parameter with a specific operation. The 2th column of each of these tables represents PSNR of the watermarked image after different attacks, 3th, 4th and 5th columns of each table represent the  $\gamma$  value of extracted watermark. The best value in each row has been bolded.



TABLE 4: Comparison of proposed method and two related methods.

Host image size	Watermark size	Average PSNR of watermarked Image			Average PSNR of extracted watermark after different attacks		
		Proposed method	Mohanty method [24]	Asatryan method [27]	Proposed method	Mohanty method [24]	Asatryan method [27]
512 × 512	256 × 256	<b>49.81</b> dB	39.44 dB	34.72 dB	<b>19.69</b> dB	16.04 dB	18.10 dB
512 × 512	128 × 128	<b>53.15</b> dB	42.31 dB	35.63 dB	<b>18.68</b> dB	15.57 dB	17.96 dB
256 × 256	128 × 128	<b>48.18</b> dB	37.98 dB	33.55 dB	<b>20.70</b> dB	16.93 dB	18.79 dB
256 × 256	64 × 64	<b>52.90</b> dB	41.09 dB	34.73 dB	<b>18.74</b> dB	15.59 dB	18.52 dB

TABLE 5: Error rates of detecting watermark.

Attack type	Proposed method	Mohanty method [24]	Asatryan method [27]
Attack-free	<b>0</b> /2500	<b>0</b> /2500	<b>0</b> /2500
Blurring (2,3)	<b>2</b> /2500	<b>2</b> /2500	<b>2</b> /2500
Sharpening(1)	<b>3</b> /2500	<b>4</b> /2500	<b>1</b> /2500
Median Filter (5 × 5, 7 × 7)	<b>0</b> /2500	<b>3</b> /2500	<b>5</b> /2500
Gaussian Noise (0.001, 0.002)	<b>3</b> /2500	<b>3</b> /2500	<b>2</b> /2500
Gaussian Low Pass Filter (3 × 3, 5 × 5)	<b>0</b> /2500	<b>0</b> /2500	<b>1</b> /2500
Cropping (40%, 50%, 60%)	<b>1</b> /2500	<b>20</b> /2500	<b>1</b> /2500
Scaling (1/2, 1/4)	<b>2</b> /2500	<b>3</b> /2500	<b>7</b> /2500
JPEG Compression (10,20,30,40)	<b>3</b> /2500	<b>6</b> /2500	<b>1</b> /2500
Tampering	<b>0</b> /2500	<b>8</b> /2500	<b>2</b> /2500
Data Block Removal	<b>0</b> /2500	<b>5</b> /2500	<b>1</b> /2500
Composite Attack	<b>3</b> /2500	<b>13</b> /2500	<b>21</b> /2500
Total	<b>16</b>	<b>67</b>	<b>44</b>

TABLE 6: PSNR value of watermarked image (512 × 512) by several methods.

Image	Watermark	Watermark size	PSNR of watermarked image		
			Mohanty method [24]	Asatryan method [27]	Proposed method
Lena	Figure 11(b)	128 × 128	43.35 dB	36.10 dB	<b>52.33</b> dB
Lena	Figure 11(b)	256 × 256	40.71 dB	36.06 dB	<b>48.27</b> dB
Lena	Figure 11(d)	128 × 128	44.46 dB	35.78 dB	<b>52.92</b> dB
Lena	Figure 11(d)	256 × 256	40.98 dB	35.82 dB	<b>48.84</b> dB
Baboon	Figure 11(b)	128 × 128	43.09 dB	36.42 dB	<b>52.44</b> dB
Baboon	Figure 11(b)	256 × 256	41.01 dB	36.46 dB	<b>48.47</b> dB
Baboon	Figure 11(d)	128 × 128	44.19 dB	35.91 dB	<b>53.08</b> dB
Baboon	Figure 11(d)	256 × 256	41.17 dB	35.92 dB	<b>49.11</b> dB
Peppers	Figure 11(b)	128 × 128	43.27 dB	35.18 dB	<b>51.72</b> dB
Peppers	Figure 11(b)	256 × 256	39.03 dB	35.20 dB	<b>47.94</b> dB
Peppers	Figure 11(d)	128 × 128	43.18 dB	34.90 dB	<b>52.27</b> dB
Peppers	Figure 11(d)	256 × 256	40.80 dB	34.86 dB	<b>48.66</b> dB
Crowd	Figure 11(b)	128 × 128	43.69 dB	35.33 dB	<b>51.38</b> dB
Crowd	Figure 11(b)	256 × 256	40.30 dB	35.37 dB	<b>47.63</b> dB
Crowd	Figure 11(d)	128 × 128	44.12 dB	35.10 dB	<b>52.04</b> dB
Crowd	Figure 11(d)	256 × 256	40.25 dB	35.04 dB	<b>48.26</b> dB



TABLE 7: The quality of extracted watermark from Lena image ( $512 \times 512$ ) with watermark size  $256 \times 256$  (Figure 11(b)) versus several attacks in different methods.

Attack type	Watermarked image PSNR	$\gamma$ of extracted watermark		
		Mohanty method [24]	Asatryan method [27]	Proposed method
No Attack	48.27 dB	0.9975	<b>0.9988</b>	0.9975
AF( $5 \times 5$ ) + JP(50)	29.49 dB	0.6533	0.6898	<b>0.7008</b>
S(1/2) + B(3) + JP(60)	28.91 dB	0.5611	0.5709	<b>0.5770</b>
SH(1) + MF( $5 \times 5$ )	29.25 dB	0.5753	<b>0.7128</b>	0.6389
MF( $5 \times 5$ ) + S(1/2) + JP(60)	30.25 dB	0.5901	0.4866	<b>0.7178</b>
GN(0,0.002) + MF( $5 \times 5$ ) + JP(50)	30.03 dB	0.4569	<b>0.5684</b>	0.5360
GL( $5 \times 5$ ) + CAR(1,1.2) + JP(50)	26.38 dB	0.7072	0.6079	<b>0.8984</b>
WF( $3 \times 3$ ) + B(2) + JP(40)	31.20 dB	0.5310	0.6404	<b>0.6696</b>
B(2) + GN(0,0.002) + JP(50)	29.38 dB	0.4085	0.4402	<b>0.5535</b>
C(40%)	18.01 dB	0.1708	0.3508	<b>0.4506</b>
S(1/4) + JP(40)	28.59 dB	0.4598	0.2222	<b>0.6341</b>

TABLE 8: The quality of extracted watermark from Baboon image ( $512 \times 512$ ) with watermark size  $256 \times 256$  (Figure 11(b)) versus several attacks in different methods.

Attack type	Watermarked image PSNR	$\gamma$ of extracted watermark		
		Mohanty method [24]	Asatryan method [27]	Proposed method
No Attack	48.47 dB	0.9961	<b>0.9986</b>	0.9966
AF( $5 \times 5$ ) + JP(50)	24.32 dB	0.3857	<b>0.5036</b>	0.4550
S(1/2) + B(2) + JP(60)	25.59 dB	0.3973	0.4318	<b>0.5109</b>
SH(1) + MF( $5 \times 5$ )	23.86 dB	0.3128	<b>0.6555</b>	0.5246
MF( $5 \times 5$ ) + S(1/2) + JP(60)	24.37 dB	0.3033	0.4202	<b>0.4293</b>
GN(0, 0.002) + MF( $5 \times 5$ ) + JP(50)	24.34 dB	0.3048	<b>0.5439</b>	0.4032
GL( $5 \times 5$ ) + CAR(1,1.2) + JP(50)	27.68 dB	0.4485	0.3643	<b>0.6176</b>
WF( $3 \times 3$ ) + B(2) + JP(40)	26.48 dB	0.4079	0.4459	<b>0.5079</b>
B(2) + GN(0,0.002) + JP(50)	25.13 dB	0.4239	<b>0.5226</b>	0.5020
C(40%)	20.29 dB	0.1279	0.3880	<b>0.4183</b>
S(1/4) + JP(40)	23.45 dB	0.2017	0.1212	<b>0.4027</b>

TABLE 9: The quality of extracted watermarked from Peppers image ( $512 \times 512$ ) with watermark size  $128 \times 128$  (Figure 11(b)) versus several attacks in different methods.

Attack type	Watermarked image PSNR	$\gamma$ of extracted watermark		
		Mohanty method [24]	Asatryan method [27]	Proposed method
No Attack	51.72 dB	0.9974	<b>0.9990</b>	0.9979
AF( $5 \times 5$ ) + JP(50)	29.80 dB	0.4233	<b>0.6118</b>	0.5943
S(1/2)+B(3) + JP(60)	29.35 dB	0.3711	0.4999	<b>0.5492</b>
SH(1) + MF( $5 \times 5$ )	29.52 dB	0.4277	<b>0.6256</b>	0.5750
MF( $5 \times 5$ ) + S(1/2) + JP(60)	30.66 dB	0.4405	0.3304	<b>0.5490</b>
GN(0, 0.002) + MF( $5 \times 5$ ) + JP(50)	30.34 dB	0.2832	<b>0.4481</b>	0.4021
GL( $5 \times 5$ ) + CAR(1,1.2) + JP(50)	31.79 dB	0.6008	0.4568	<b>0.7637</b>
WF( $3 \times 3$ ) + B(2) + JP(40)	31.13 dB	0.5343	0.5928	<b>0.6697</b>
B(2) + GN(0,0.002) + JP(50)	27.60 dB	0.3900	<b>0.5222</b>	0.5152
C(40%)	17.06 dB	0.2080	0.3710	<b>0.6544</b>
S(1/4) + JP(40)	29.06 dB	0.3827	0.3535	<b>0.6415</b>

TABLE 10: The quality of extracted watermark from Crowd image ( $512 \times 512$ ) with watermark size  $128 \times 128$  (Figure 11(b)) versus several attacks in different methods.

Attack type	Watermarked image PSNR	y of extracted watermark		
		Mohanty method [24]	Asatryan method [27]	Proposed method
No Attack	51.38 dB	0.9969	<b>0.9985</b>	0.9974
AF( $5 \times 5$ ) + JP(50)	25.19 dB	0.3432	<b>0.4803</b>	0.4346
S(1/2) + B(3) + JP(60)	24.30 dB	0.3040	<b>0.4584</b>	0.4490
SH(1) + MF( $5 \times 5$ )	24.53 dB	0.2400	<b>0.4056</b>	0.3132
MF( $5 \times 5$ ) + S(1/2) + JP(60)	25.79 dB	0.3029	0.3505	<b>0.4132</b>
GN(0, 0.002) + MF( $5 \times 5$ ) + JP(50)	25.90 dB	0.2306	<b>0.4034</b>	0.3389
GL( $5 \times 5$ ) + CAR(1,1.2) + JP(50)	29.24 dB	0.4064	0.4708	0.5851
WF( $3 \times 3$ ) + B(2) + JP(40)	27.91 dB	0.3666	0.4660	<b>0.5051</b>
B(2) + GN(0,0.002) + JP(50)	26.06 dB	0.3099	<b>0.5306</b>	0.4455
C(40%)	14.70 dB	0.1818	0.3195	<b>0.7337</b>
S(1/4) + JP(40)	23.86 dB	0.2030	0.2344	<b>0.4182</b>

## 5. Conclusion

In this paper, a grayscale watermark insertion and extraction schemes were proposed. The proposed method works by modifying the DC value of the original image in frequency domain to create the watermarked image. The embedding procedure is based on fuzzy inference system to locate the best place of watermark insertion. The algorithm was tested with several standard test images and the experimental results demonstrated that it created high-quality images and it was robust versus different attacks. In the future, we are going to change the proposed method such that it can support all attacks by developing a blind method that uses similar idea.

## References

- [1] M.-C. Hu, D.-C. Lou, and M.-C. Chang, "Dual-wrapped digital watermarking scheme for image copyright protection," *Computers and Security*, vol. 26, no. 4, pp. 319–330, 2007.
- [2] J.-M. Shieh, D.-C. Lou, and M.-C. Chang, "A semi-blind digital watermarking scheme based on singular value decomposition," *Computer Standards and Interfaces*, vol. 28, no. 4, pp. 428–440, 2006.
- [3] C. A. Gonzales, L. Allman, T. McCarthy, P. Wendt, and A. N. Akansu, "DCT coding for motion video storage using adaptive arithmetic coding," *Signal Processing*, vol. 2, no. 2, pp. 145–154, 1990.
- [4] K. Veeraswamy and S. Srinivas Kumar, "Adaptive AC-coefficient prediction for image compression and blind watermarking," *Journal of Multimedia*, vol. 3, no. 1, pp. 16–22, 2008.
- [5] V. Martin, M. Chabert, and B. Lacaze, "An interpolation-based watermarking scheme," *Signal Processing*, vol. 88, no. 3, pp. 539–557, 2008.
- [6] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–335, 1996.
- [7] R. V. Schyndel, A. Z. Trikel, and C. F. Osborn, "A digital watermark," in *Proceedings of the 1st IEEE International Conference on Image Processing*, pp. 86–90, 1994.
- [8] V. Saxena and J. P. Gupta, "A novel watermarking scheme for JPEG images," *WSEAS Transactions on Signal Processing*, vol. 5, no. 2, pp. 74–84, 2009.
- [9] X. Kang, W. Zeng, and J. Huang, "A multi-band wavelet watermarking scheme," *International Journal of Network Security*, vol. 6, no. 2, pp. 121–126, 2008.
- [10] X.-Y. Wang, L.-M. Hou, and J. Wu, "A feature-based robust digital image watermarking against geometric attacks," *Image and Vision Computing*, vol. 26, no. 7, pp. 980–989, 2008.
- [11] Z.-J. Lee, S.-W. Lin, S.-F. Su, and C.-Y. Lin, "A hybrid watermarking technique applied to digital images," *Applied Soft Computing*, vol. 8, no. 1, pp. 798–808, 2008.
- [12] A. K. Parthasarathy and S. Kak, "An improved method of content based image watermarking," *IEEE Transactions on Broadcasting*, vol. 53, no. 2, pp. 468–479, 2007.
- [13] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 148–158, 2001.
- [14] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shanon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [15] C. S. Lu, H. Y. M. Liao, and C. J. Sze, "Cocktail watermarking on images," in *Proceedings of the 3rd International Workshop on Information Hiding*, pp. 333–347, 1999.
- [16] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing*, vol. 66, no. 3, pp. 357–372, 1999.
- [17] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, 1999.
- [18] J. Huang, Q. S. Yun, and W. Cheng, "Image watermarking in DCT: an embedding strategy and algorithm," *Acta Electronica Sinica*, vol. 28, no. 4, pp. 57–60, 2000.
- [19] L. Xie and G. R. Arce, "Joint wavelet compression and authentication watermarking," in *Proceedings of the 1998 International Conference on Image Processing*, pp. 427–431, October 1998.
- [20] Y. Zhao, P. Campisi, and D. Kundur, "Dual domain watermarking for authentication and compression of cultural heritage images," *IEEE Transactions on Image Processing*, vol. 13, no. 3, pp. 430–448, 2004.

- [21] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, 1999.
- [22] C.-T. Hsu and J.-L. Wu, "Multiresolution watermarking for digital images," *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 8, pp. 1097–1101, 1998.
- [23] A. Al-Haj, "Combined DWT-DCT digital image watermarking," *Journal of Computer Science*, vol. 3, no. 9, pp. 740–746, 2007.
- [24] S. P. Mohanty and B. K. Bhargava, "Invisible watermarking based on creation and robust insertion-extraction of image adaptive watermarks," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 5, no. 2, article 12, 2008.
- [25] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 835–839, 1983.
- [26] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A dual watermarking technique for images," in *Proceedings of the 7th ACM International Multimedia Conference*, pp. 49–51, 1999.
- [27] D. Asatryan and N. Asatryan, "Combined spatial and frequency domain watermarking," in *Proceedings of the 7th International Conference on Computer Science and Information Technologies*, pp. 323–326, 2009.
- [28] N. Sakr, J. Zhao, and V. Z. Groza, "Adaptive digital image watermarking based on predictive embedding and a Dynamic Fuzzy Inference System model," *International Journal of Advanced Media and Communication*, vol. 1, no. 3, pp. 237–264, 2007.
- [29] A. G. Borş and I. Pitas, "Image watermarking using block site selection and DCT domain constraints," *Optics Express*, vol. 3, no. 12, pp. 512–523, 1998.