

# Adaptive image watermarking using human perception based fuzzy inference system

Maedeh Jamali<sup>a</sup>, Shima Rafiei<sup>a</sup>, S.M. Reza Soroushmehr<sup>b,\*</sup>, Nader Karimi<sup>a</sup>, Shahram Shirani<sup>c</sup>, Kayvan Najarian<sup>b</sup> and Shadrokh Samavi<sup>a,b,c</sup>

<sup>a</sup>*Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran*

<sup>b</sup>*Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, USA*

<sup>c</sup>*Department of Electrical and Computer Engineering, McMaster University, Hamilton, L8S 4L8, Canada*

**Abstract.** Development of digital content has increased the necessity of copyright protection using watermarking. Imperceptibility and robustness are two important features of watermarking algorithms. The goal of watermarking methods is to satisfy the tradeoff between these two contradicting characteristics. Recently, watermarking methods in transform domains have displayed favorable results. In this paper, we present an adaptive blind watermarking method, which has high imperceptibility in areas that are important to the human visual system. We propose a fuzzy system to control the embedding strength factor adaptively. Image saliency, intensity, and edge-concentration are shown to be important to a human observer and are hence used as fuzzy attributes. Embedding is performed in the discrete cosine transform of the wavelet domain to achieve high imperceptibility and acceptable robustness. Experimental results show the superiority of the proposed algorithm over comparable methods.

**Keywords:** Blind watermarking, fuzzy system, saliency, discrete wavelet transform, adaptive strength factor

## 1. Introduction

Digital imaging has developed quickly in the last two decades. Because of easy access to the internet, sharing digital media has become easier and faster in recent years. Unlike their analog counterparts, digital contents can be copied with the same quality as the original one. The ease of replication, along with the access to image editing tools, the integrity of digital media can be threatened. Increasing the possibility of digital tampering has emphasized the need for sophisticated techniques of copyright protection and prevention of unauthorized copying and distribution [1]. Watermarking is one of the most common and trusty ways to solve this problem. In watermarking methods, the owner's copyright infor-

mation is embedded into an image that later can be extracted from the embedded image [2]. Three main characteristics of watermarking are robustness, imperceptibility, and capacity [3]. The efficiency of a watermarking technique relies highly on the trade-off among these features. A watermarking method should be robust against adversarial attacks, and the embedded logo should be correctly extracted from the watermarked image. In addition, the embedding process should be as transparent as possible, and the watermarked image should be as similar to the cover image as possible. The watermarking algorithm should also provide enough capacity for embedding of the desired watermark logo.

The watermarking methods can be classified in different ways based on the embedding domain, the embedding method, and the extraction process. Regarding the extraction process, watermark algorithms could be classified into three categories. A *blind* watermarking method is the one that does not

\*Corresponding author. S.M. Reza Soroushmehr, Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, 48109, USA. E-mail: ssoroush@umich.edu.

require the cover image for the extraction of the logo. Hence, a *non-blind* method is the one that needs the cover image for the extraction process. *Semi-blind* methods require some metadata alongside the watermarked image during the extraction phase. Hence, blind methods are superior to the other two extraction approaches. An approach, besides watermarking, is the hashing of the original image. If the hash output, on the receiver side, is the same as the one transmitted from the original image, then the image is considered as unchanged [4, 5]. However, such hashing methods require a secure channel for image transmission. Due to the rare availability of such requirements, watermarking is the prevalent means of copyright protection.

Another type of classification for watermark methods is based on the embedding domain. Watermarking can be categorized into two groups of spatial and transform domain techniques. In spatial domain methods, information is directly embedded in image pixels. However, these methods usually are not necessarily robust against usual image processing attacks. On the other hand, frequency domain algorithms are more robust in most cases [3].

In this paper, we propose a new adaptive blind watermarking based on human visual perception. To achieve a tradeoff between imperceptibility and robustness, we use adaptive control of the embedding strength factors calculated by a fuzzy system. The fuzzy system models the uncertainty and inference similar to a human observer. Saliency, edge-concentration, and intensity are three fuzzy inputs that form the fuzzy term set. After calculating the strength factor based on the fuzzy system, the embedding is performed.

In brief, we propose a novel method using a fuzzy system for finding regions where the human visual system (HVS) is not sensitive to their changes. Embedding is done with different strengths in different wavelet sub-bands.

The rest of this paper is organized as follows. Section II is a review of some existing watermarking methods. Section III contains the details of the proposed method. In Section IV the experimental results are presented, and finally, the paper is concluded in Section V.

## 2. Literature review

Embedding domain in watermarking methods can be either spatial or transform. Some spatial domain

methods are proposed in [6–8]. Lin et al. [6] proposed watermarking scheme using the  $1/T$  rate *forward error correction* (FEC) where  $T$  is the data redundancy rate. For better security, the watermark logo is mixed with noise bits. Also, they XOR the watermark with a binary feature value of the image by  $1/T$  rate FEC. Their algorithm has a blind extraction phase and the watermark bits are determined by majority voting. In [7], the embedding phase starts by preprocessing of the cover image by a Gaussian low-pass filter and using a secret key, a number of gray levels are randomly selected. A histogram of the filtered image based on these selected gray levels is made. The novelty of their work is using the histogram-shape-related index to select pixel groups with the highest number of pixels. Also, a safe band is constructed between chosen and non-chosen pixel groups. The watermark is then inserted into the chosen pixel groups. However, the capacity of this method is low. Work of [8] is another spatial domain technique that a set of affine invariants is derived from Legendre moments. These affine invariants are used in embedding.

Transform methods embed information into frequency coefficients of the original image. Hence, they can be more powerful than the spatial domain groups and better preserve image imperceptibility [5]. Many different transform techniques, such as *discrete fourier transform* (DFT) [9], *discrete cosine transform* (DCT) [10–18], *discrete wavelet transform* (DWT) [19–32] and *Contourlet transform* (CT) [33–40] have been used for digital image watermarking. In [9], the watermark is added in middle frequencies of DFT, and it has circular symmetry. Suhail and Obaidat [10] propose a method in which an input image is divided into different parts based on the Voronoi algorithm. After that, the DCT of these segmented parts hides a sequence of real numbers. Authors of [11] apply DCT to image blocks, and a low-resolution approximate image is formed using the DC coefficient of each block. Eventually, embedding is done by adding watermark to high frequencies of the reconstructed image. Authors of [12] propose a DCT based method where the input image is divided into  $8 \times$  blocks. Then block is transferred into DCT and some blocks based on a Gaussian network classifier are selected. Finally, DCT coefficients are modified based on the watermark data. Heidari et al. [13] proposes a watermarking method and show that different attacks tend to change different parts of the frequency spectrum. Hence, by detecting how different attacks affect the image, one can perform a more precise extraction. They propose an attack classification method to

recognize regions of the frequency domain that are less damaged. Hence, the watermark can be extracted from regions with less damage. Huang and Guan proposed a hybrid DCT and *singular value decomposition* (SVD) based watermarking [14]. In their method, SVD and DCT are applied to the watermark and original images respectively, and the singular values of the watermark are embedded into DCT coefficients of the original image. Another DCT-SVD based algorithm is proposed in [15]. First, a mask of the original image is built using a luminance mask. The embedding process is done by modifying singular values of DCT of the original image with singular values of a produced mask. Genetic algorithm finds the control parameter. In [16], an adaptive watermarking algorithm for medical images is proposed which uses *fuzzy inference system* (FIS) and characteristics of the HVS. The embedding is done in the wavelet domain. The HVS model contains luminance, texture, and frequency sensitivities. The algorithm of [17] uses a neuro-fuzzy method consisting of back propagation neural networks and fuzzy logic techniques for embedding and extraction procedures. In their algorithm, HVS parameters are fed into the system, and the output of fuzzy-neural is used as the strength-factor for the watermark embedding. In [18], three fuzzy inference models are used for formation of the watermark strength factor, which uses inputs based on HVS.

A survey of DWT base watermarking is explained in [19]. DWT is widely used in watermarking algorithms. The middle or high-frequency regions of the coefficients are usually used for embedding [20]. Rasti et al. [21] introduce a watermarking method based on DWT. They add pseudo-random codes to large coefficients at high and middle-frequency bands, but the high-frequency band is not robust against attacks such as JPEG. In [22], a DWT-SVD method is introduced. They embed a watermark in singular values of the wavelet transform's sub-bands of the original image. Authors of [23] propose wavelet tree clustering for data hiding. Distance vector produces wavelet trees. These trees are classified into two clusters: one denotes a watermark bit of 1 and the other shows 0. Statistical difference and the distance vector of a wavelet tree are compared for the extraction of the embedded bit. In [24], a blind watermarking method is proposed which uses the quantization of the maximum wavelet coefficient. Wavelet coefficients of the input image are categorized into different blocks. The embedding is done in different sub-bands. They add various energies to

maximum coefficients so that, the chosen coefficient remains maximum in that given block. This method has the drawback of having a low normalized correlation (NC) against intense JPEG attacks. In [25, 26] the combination of DCT and DWT domains is used. The watermark is embedded in the mid-frequency coefficients in the DCT domain of three DWT levels of the LL band of the original image. In some research works, a combination of DWT and SVD is used in watermarking. Authors of [27] apply SVD to all frequency bands in DWT of the original image for the watermarking purpose. In [28], a quantization-based watermarking is proposed. They embed a watermark bit by quantizing the angles of considerable gradient vectors in different wavelet scales. Authors of [29] have proposed a geometrical model for embedding to generate a tradeoff between robustness and imperceptibility. They used eight samples of wavelet approximation coefficients from each image block and built two line-segments in a two-dimensional space. The proposed method of [30] is based on context modeling and fuzzy inference filter, which are used to determine coefficients with large entropy in coarser DWT sub-bands for watermark embedding. The algorithm proposed in [31] utilizes fuzzy logic to obtain a perceptual weighting factor for each wavelet coefficient for the embedding at different scales of an image.

Some researchers have used different types of transforms such as CT, DCT, and Hadamard transform for the embedding process. Ghannam et al. [33] proposed to embed a watermark into different bands of CT to increase the robustness. Authors of [34] introduce an adaptive blind watermarking in which the watermark is embedded in DCT coefficients of CT. They apply two-level CT to the cover image. In the first level, the approximate image is divided into blocks. Important edges of each block are determined using their proposed edge detection method. Parts of an image with a high concentration of edges are considered as candidate parts for strong embedding. Some portions of the second level are also concatenating with mentioned blocks. The entropy of blocks and some other criterions of each block determine an adaptive strength factor for that block. Then DCT transform of blocks is used for embedding. Authors of [35] use Hadamard transform for obtaining robust and low complexity embedding mechanism. Kaviani et al. [36], use a hybrid of CT and DCT for embedding environment. They use the CT complexity of each block to set the strength factor of each block. In [37], the CT domain is used for embedding. They embed the

watermark in DCT coefficients of CT blocks for more robustness. A major drawback of this method is its poor performance against attacks such as salt & pepper noise, Gaussian noise, and JPEG compression. The method of [38] uses maximum likelihood method based on normal inverse Gaussian (NIG) distribution for watermark extraction. Authors of [39] have proposed a watermarking method based on a sample projection technique. They use low-frequency components of image blocks for embedding to achieve more robustness against different attacks. They use four samples of approximation coefficients of image blocks to build line-segment in 2D space. The slope of this line segment is used for embedding. Authors of [40] propose a Contourlet-based watermarking method. They use nine samples of approximation coefficients of an image block to build a plane in 3D space. Embedding is done by changing the dihedral angle between the created plane and the x-y plane.

Fuzzy logic models have been used in watermarking algorithms to obtain appropriate weighting factors to embed watermarks. Most FIS and HVS-based algorithms in the literature heuristically find membership functions as their input features. The membership functions are not tuned subjectively by users or objectively by the real distributions of the features. Furthermore, none of the mentioned algorithms considers saliency, based on human fixation locations, which could be an effective FIS input.

### 3. Proposed method

In this Section, we explain our proposed watermarking method that has three main steps; feature extraction, fuzzy system, and watermarking scheme. As our goal is to design a robust and imperceptible watermark method, we extract features based on HVS. In this method, the output of the first step is fed to a fuzzy system to produce a probability map showing the importance of different parts of the image according to input characteristics. The output of the fuzzy system is a map that acts as an adaptive strength factor for the watermarking system. We first explain our feature extraction phase and how the fuzzy system inputs are prepared. After that, each fuzzy model, the embedding, and extraction phases are explained.

#### 3.1. Features extraction

Here we consider saliency, edge concentration, and intensity features that could improve the embedding

robustness. Therefore, we look for regions where the embedding has least perceptual effects. The first part of Fig. 1 shows the feature extraction phase. In the following, we explain those features.

##### 3.1.1. Saliency

The main idea in saliency detection algorithms is to identify the most significant and visually informative parts of a scene. Salient parts are supposed to indicate human fixation locations of an image. Visual saliency is a significant characteristic of HVS that tries to predict the most relevant and important regions of images viewed by the human eyes. Our eyes receive lots of information from its surrounding, but our brains cannot process all of them. Therefore, the brain tries to process the most important information selected based on the visual features of the stimulus [48]. Scientists have tried to model the human attention and saliency in a computerized form that leads to saliency detection methods. Usually, saliency detection methods produce a saliency map where salient parts have more intensity than other regions.

To achieve imperceptible watermark, it is desirable to perform embedding in parts of an image that human eyes are less sensitive to their changes. Also, stronger data hiding in non-salient parts helps robustness. A large number of saliency approaches with a different mechanism for salient part detection have been proposed. These approaches can be categorized into two groups of *top-down* and *bottom-up* methods. Top-down methods usually use high-level information as prior knowledge and are task dependent. Bottom-up methods, on the other hand, consider saliency detection as a high-level task, and in addition to background, the foreground dependency is considered [49]. The method of [41] is a bottom-up method that consists of two steps: forming activation maps on certain feature channels, and then normalizing them in a way that highlights conspicuity and admits combination with other maps. Its output is a map with values in the range of [0, 1]. The closer the map values are to 1, the more salient the corresponding areas would be. The saliency map implicitly indicates that changes in such salient areas attract more attention of the viewers. On the other hand, regions with map values closer to 0 are less important and more suitable for strong embedding. Therefore, saliency shows the importance of a region and can be used to control the strength of watermark embedding. The saliency map is fed into the fuzzy system for further processing.

Figure 2(a) shows the original image. Figure 2(b) shows the saliency map with values in the range of

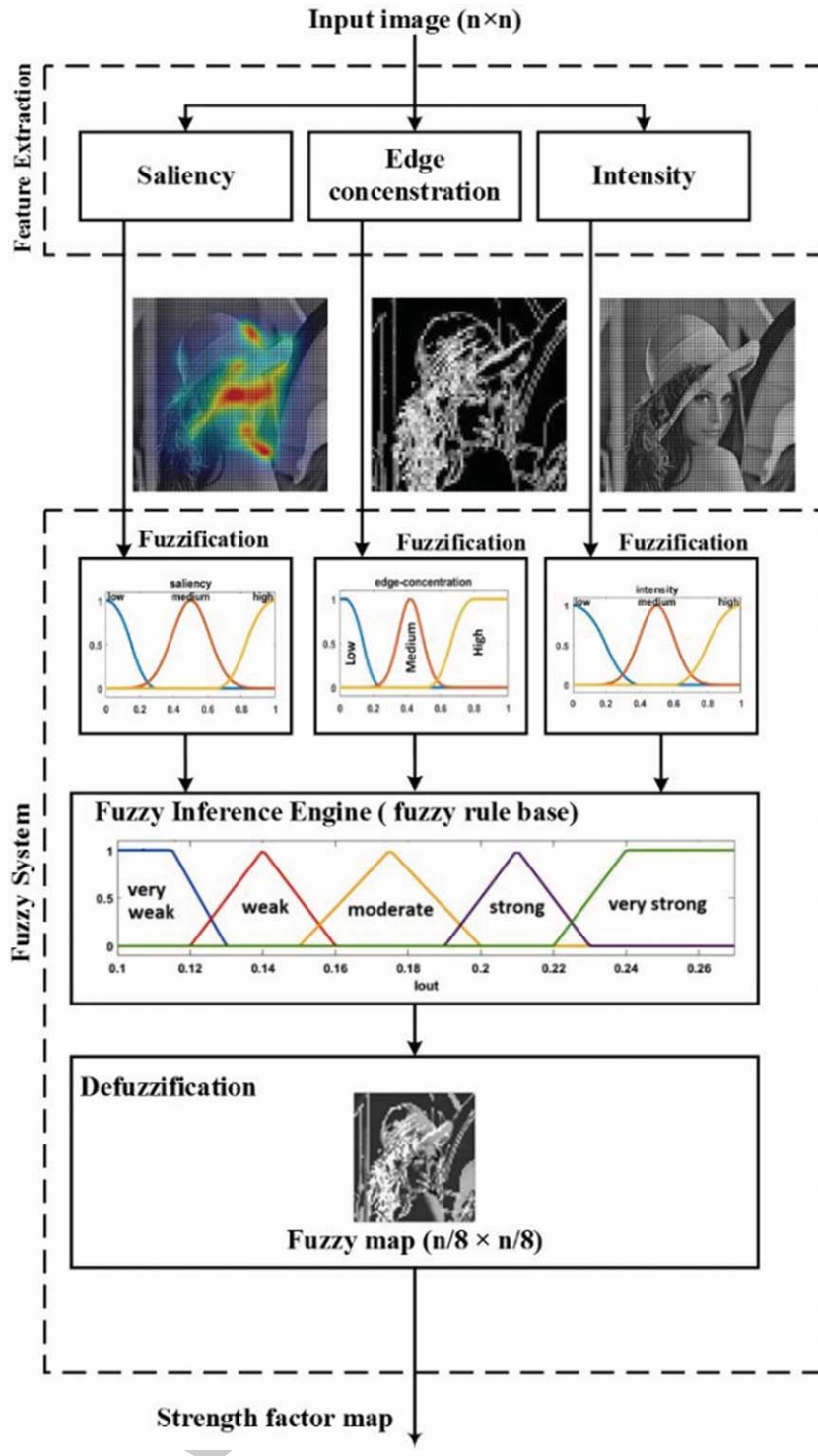


Fig. 1. Block diagram of the proposed feature extraction and strength-factor map formation.

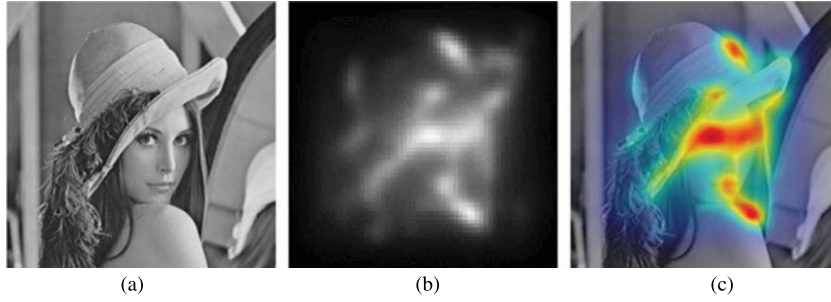


Fig. 2. Saliency map (a) original image, (b) output of saliency method [41], (c) heat map of saliency.

[0,1], which are calculated using the method of Harel et al. [41]. Also, in Fig. 2(c) a heat map of the measured saliency values is shown. In the saliency map, warmer colors (red to orange) present more salient regions whereas cooler colors (blue to gray) are for less important parts. This heat map is shown only to convey a better impression of the saliency map.

### 3.1.2. Intensity

The just-noticeable-difference (JND) is the minimum amount of change that a person can detect at least 50% of the time. Visual JND is called Weber ratio, and it is expressed as  $\Delta I/I$  where  $I$  is the intensity [50]. For a region with high intensity,  $\Delta I$  has a higher value as compared to a region with low intensity. Therefore, changes in regions with high intensities could be undetectable by a human as opposed to dark areas.

We used a simple watermark method on Lena and showed the output in Fig. 3. This image has high and low-intensity regions, next to each other, which facilitates an intuitive comparison between these areas before and after embedding. Figure 3(c) shows the watermarked image and Fig. 3(d) shows a zoomed area of Fig. 3(c). There is negligible change in the high-intensity area while texture noise is visible in dark areas. Hence, we try to have more powerful embedding in regions with high intensities as opposed to other regions.

Here, we choose pixel intensity as an HVS based feature for adaptive embedding in different parts of an image. For this aim, we first partition an image into  $8 \times 8$  blocks and calculate the average intensity of each block. For an image with the size of  $512 \times 512$ , the average block intensity values make a  $64 \times 64$  intensity map. We normalize this map to [0, 1] and select higher intensity regions (close to 1 values) as candidates for powerful embedding.

### 3.1.3. Edge-concentration

The human visual system usually cannot recognize changes in irregular parts of an image. Hence, such regions would be proper for powerful embedding. We choose regions with high concentration of edges as candidate areas for this aim. Figure 4 shows the output of the watermarking method, which uses a constant strength factor for the Lena image. We used DCT of DWT domain for watermark embedding. One smooth area is compared with the result of embedding in a cluttered region. Figure 4 (a) shows parts of the original image and Fig. 4 (b) illustrates those parts after embedding. The difference between smooth regions, before and after embedding, is noticeable. Hence, we use cluttered regions for more powerful embedding.

To identify regions with irregularity, we first use the Canny edge detector to find edges of the image. Then, regions with high edge density are selected. After that, we divide this maximum value by the obtained concentration-value of a block. Figure 5 shows the output of this phase. As can be seen in Fig. 5, regions with more edges have higher concentration values. Such blocks are more suitable for embedding purposes. Figure 5(c) is the resized version of the concentration map to illustrate it with a better resolution.

The process of edge-concentration is as follows:

1. Apply canny edge detector on the image.
2. Divide canny output into  $8 \times 8$  blocks.
3. For every position in a block find the variance of edge pixels around it in a  $3 \times 3$  neighborhood.
4. Assign to each block the average of all variance values inside of that block.
5. Normalize this value to [0, 1].

## 3.2. Fuzzy system

Fuzzy inference system (FIS) is a method that assigns an output vector to each input vector, based



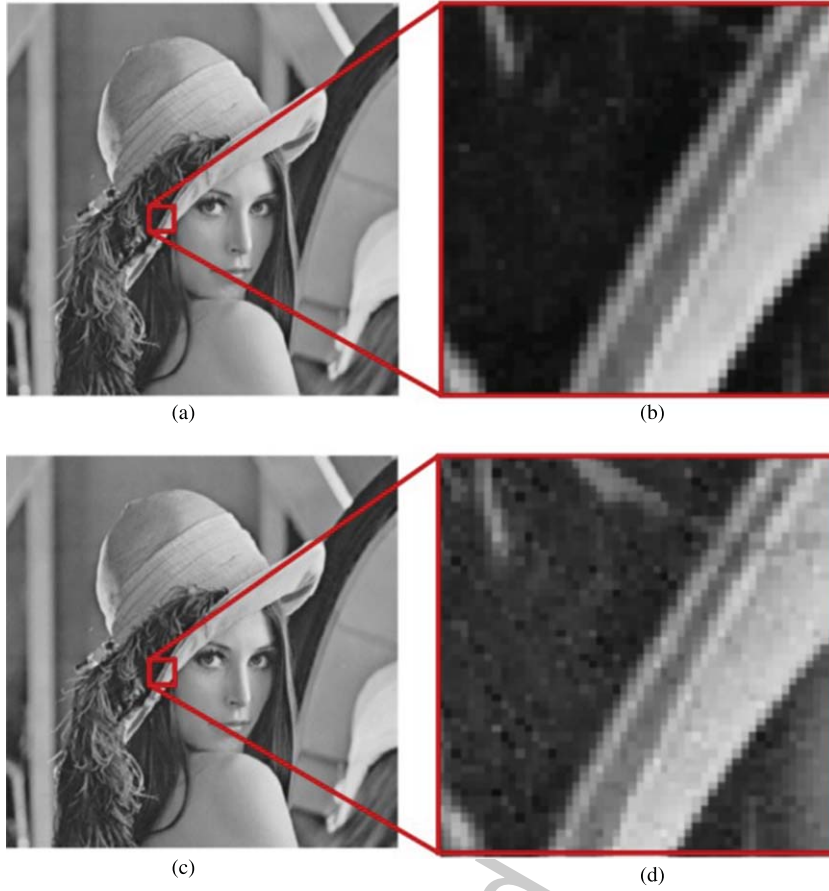


Fig. 3. Image quality comparison between high and low-intensity regions after embedding, (a) original image, (b) zoomed area of the original image (c) watermarked image, (d) zoomed area of the watermarked image

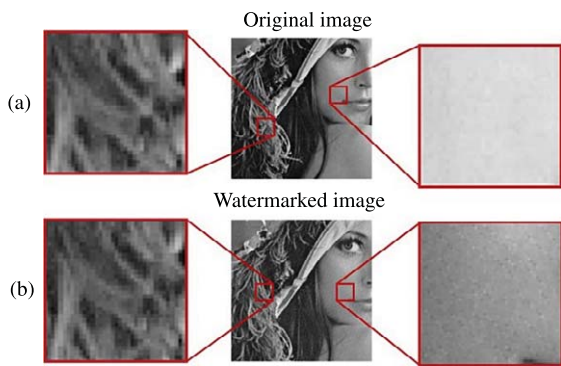


Fig. 4. Smooth and cluttered regions before and after embedding, (a) original image, (b) watermarked image.

on a set of predefined rules [42]. These rules are a list of if-then statements inspired by the human experience. These rules can efficiently make decisions like an expert [43]. In this method, we use the Mamdani min-max fuzzy inference [44]. Saliency,

intensity, and edge-concentration values are fed to FIS as input attributes. Saliency, edge concentration, and intensity features are three fuzzy inputs that produce  $64 \times 64$  matrices as inputs of the fuzzy system. For these inputs, we determine three fuzzy term sets named low, high and medium with their special membership functions. These term sets are selected based on the importance of areas for embedding. The membership functions (MF), can be any function that maps each input space (referred to as the universe of discourse) to any value in the interval of  $[0, 1]$  known as membership value and presents the degree of truth [42].

Most of the times, the choice of membership function depends on the problem. The membership function is determined heuristically, subjectively, or objectively. Heuristic methods use predefined shapes for membership functions. Since these methods are chosen to fit a given problem, they work well only for intended problems. The shape of heuristic

membership functions is not flexible enough to model all kinds of data, and selected membership functions may not reflect the actual data distribution. Therefore, we do not use a heuristic approach. We try to design membership functions of the edge-concentration attribute objectively with a clustering method. This is done by generating probability density functions for each cluster to obtain the final membership functions. Moreover, membership functions for intensity and saliency attributes have been designed subjectively using observers.

### 3.2.1. Edge-concentration membership function

For this input, designing the membership function is objectively performed based on data distribution. Since we are introducing a new edge-concentration feature, it would be difficult to design a set of subjective membership functions. This is due to the lack of understanding of proper fuzzy intervals. For this aim, a clustering algorithm can be applied to estimate the actual data distribution. Finally, the resulting clusters can be used to produce the membership functions that will properly interpret the data.

To determine the distribution of data on the edge-concentration feature, we first cluster this feature using Fuzzy C-Means (FCM) clustering [45]. For clustering we considered 9 centers, experimentally 3 times more than our term-sets, seems to split the universe of discourse more densely. In Fig. 6(a), we fit a probability distribution to each of these clusters. Then we start to merge highly overlapped distributions that are next to each other. This merging is done by considering the balance of data in each of the three newly merged distributions. Finally, we only need three probability density functions (PDF) to determine the mean ( $\mu$ ) and variance ( $\sigma^2$ ) of each of these fuzzy sets. This designing of the membership function is more suitable for data partitioning. Figure 6(b) shows membership functions obtained from the 3 newly merged distributions. The average fuzzy-set is a Gaussian function with the exact mean and variance of the related PDF. In other words, the related PDF has been scaled to become a normalized membership function with a maximum height of 1.

For the other two fuzzy-sets, i.e., high and low, the S-shaped and Z-shaped membership functions have been designed, respectively. To design these terminal functions, the peak-point and a bottom-point are needed. In our final membership functions, the peak point is the mean point of the related PDF, and the bottom point is chosen according to the variance the

distribution. In other words, we consider a Gaussian function of which one side stays at the peak level.

### 3.2.2. Saliency membership functions

Three classes of regions are considered in the saliency heat map. Regions marked as red to orange are high, regions with orange to blue are medium, and regions with blue to gray colors are low saliency regions. An expert selects each region by placing a box on each region. This procedure is performed with several randomly selected images. Probability density function (PDF) of the heat map elements of each box is calculated. We use the normalized PDF as a membership function. Ultimately, we will have three fuzzy sets that are names as high, medium, and low which they respectively have S-shaped, Gaussian, and the Z-shaped membership functions.

### 3.2.3. Intensity membership functions

Intensity membership functions are designed completely subjectively based on structural similarity (SSIM) results after embedding. SSIM is dependent on the embedding strength factor.

The SSIM value has smaller decay in blocks with high intensities. The SSIM criterion complies with the human visual system. Hence, volunteers chose membership functions. They decide on low, medium, and high-intensity values. The selection of membership function should cause the least change in SSIM values. The final intensity membership function is indicated in Fig. 8(c).

### 3.2.4. Output membership functions

The fuzzy output map has five term-set consisting of very strong, strong, moderate, weak and very weak.

Volunteers design all five fuzzy sets subjectively. Figure 8 shows the final output of the fuzzy membership function. The fuzzy membership function is built based on fuzzy rules that are explained later.

### 3.2.5. Fuzzy inference rules

For the watermarking purpose, we design fuzzy rules based on three mentioned features of saliency, edge-concentration, and intensity, to be applied to different parts of an image. We consider high, medium, and low saliency areas. We have three attributes, and each one has three values. Base on the combination of attributes and their values, we will have 27 rules that are subjectively generated by volunteers. Table 1 shows these FIS rules.



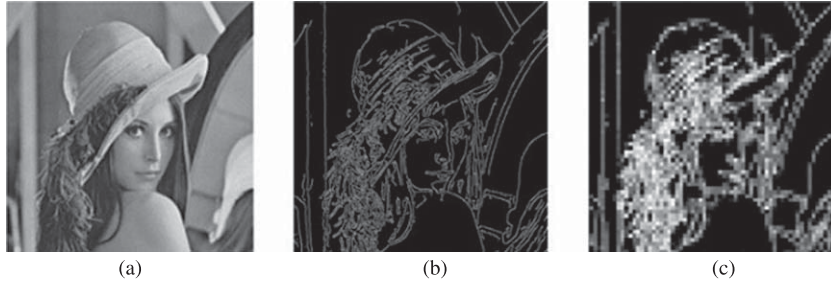


Fig. 5. Edge-concentration computation, (a) original image, (b) canny output map, (c) normalized edge-concentration map. This map is resized to  $512 \times 512$ .

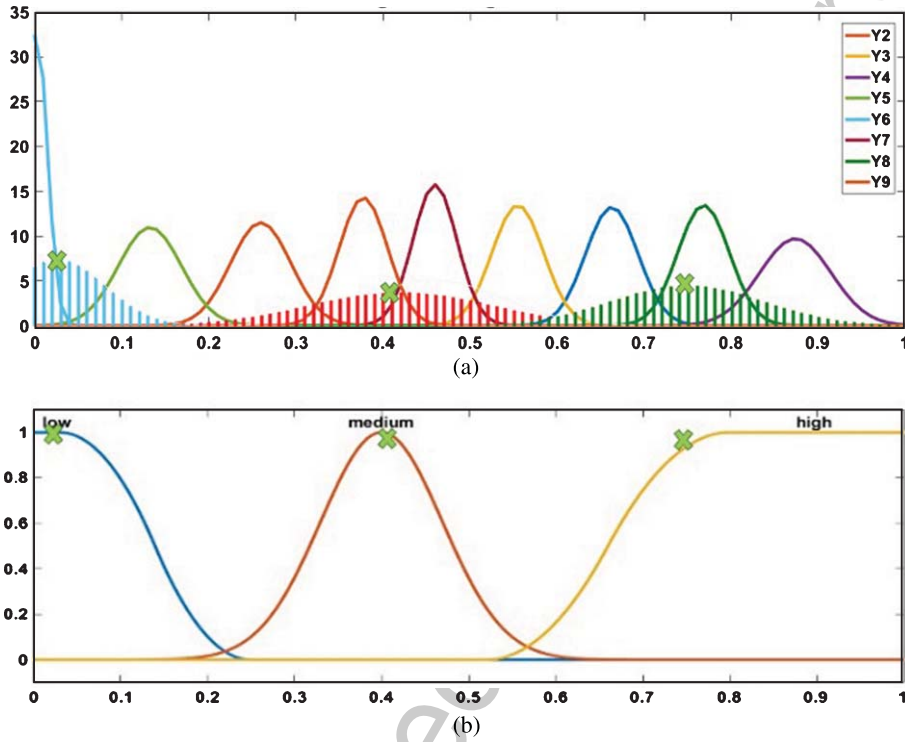


Fig. 6. Design of membership functions using FCM. (a) PDFs are fitted on 9 clusters, (b) final membership functions.

### 3.2.6. Output map

All watermark images in this study have the size of  $512 \times 512$ . Hence, there are  $64 \times 64$  non-overlapped blocks with the size of  $8 \times 8$  pixels. The defuzzified output of FIS is  $64 \times 64$  image map. In the embedding phase, we use DCT of wavelet for data hiding. Details of the embedding process are explained in the next sub-section. Hence, each pixel of the fuzzy map determines the strength factor for its corresponding  $8 \times 8$  block in DCT. Each pixel value of the fuzzy map is in the range of  $[0.1, 0.27]$ . This map is considered as the strength factor  $\alpha$  that means, blocks with unsuitable conditions, which have low intensity, low

edge-concentration, and high saliency, are assigned with the least strength factor of 0.1. This would be an initial strength factor and may be modified by a coefficient. Actually, FIS customizes the strength factor of each block using a fuzzy policy and by means of saliency, intensity, and crowdedness of pixels in a block, based on the sensitivity of the human visual system.

Figure 9 shows the output map that strength factor outputs are according to the potential of each block for accepting a strength factor in watermark embedding. The lighter pixels are the areas where stronger embedding will be done. For example, the

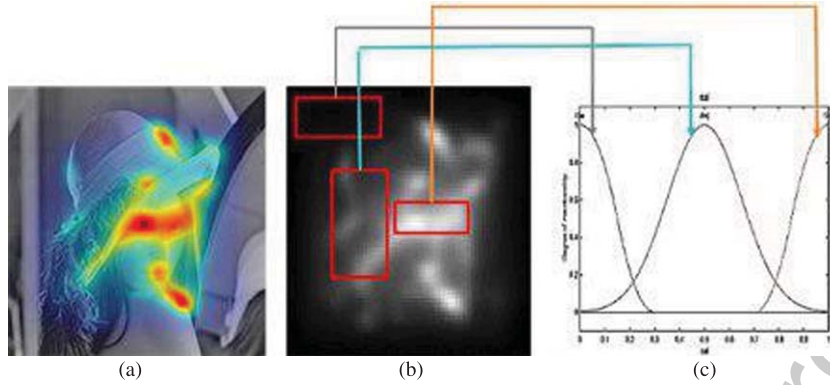


Fig. 7. Saliency membership function formation, (a) saliency heat map, (b) saliency map, (c) membership function.

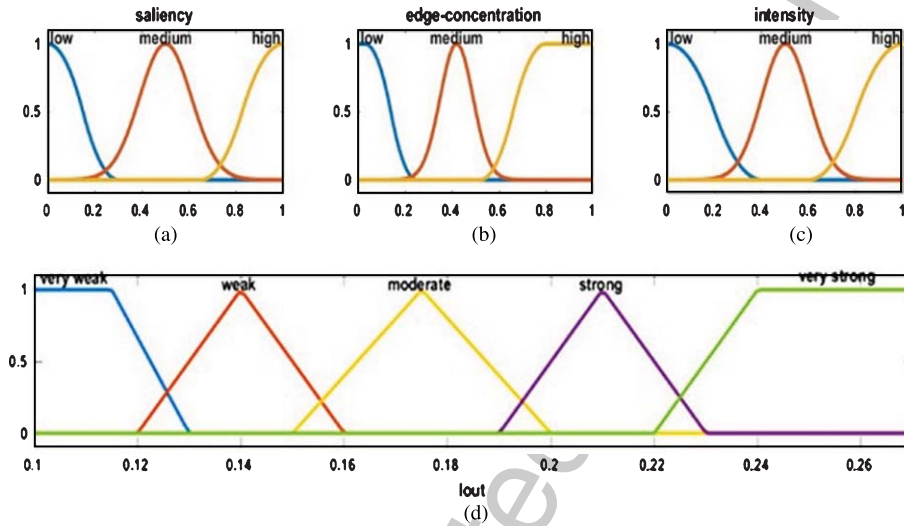


Fig. 8. Membership functions: (a) Saliency membership function, (b) edge-concentration membership function, and (c) intensity membership function. (d) Output fuzzy membership function.

fur in Lena's hat has high edge-concentration and medium saliency. This makes it suitable for very high strength factor ( $\sim 0.27$ ). Although Lena's mouth and eyes have high edge-concentration, human eyes first detect high saliency regions. Hence, the embedding factor for Lena's mouth and eyes should be lower than the fur parts of the hat. Figure 9(c) shows the output of our method and Fig. 9(d) illustrates the output of a simple watermarking method that uses constant strength factor. As can be seen, our method has considerably better outputs.

### 3.3. Watermarking scheme

Embedding in the spatial domain is very vulnerable to attack, and the embedded data is easily lost. Hence, spatial domain is not very suitable for data

embedding, and we use transform domain instead. Various transform domains have been used such as DCT [10–18], DWT [19–32] and DFT [9]. DWT is one of the preferred domains that divides an image into different frequency levels. This division allows us to embed the data in different levels redundantly and hence improve the robustness of our method. For this aim, we use a combination of DWT and DCT domains. DCT of blocks in each DWT sub-band are considered for embedding. We use HL, LH and HH sub-bands of a 2D wavelet transform in two levels. Figure 10 shows the selected sub-bands of the wavelet domain with different colors.

Due to the high sensitivity of image imperceptibility to changes in LL coefficients, we do not use LL sub-band for embedding purpose. Also, the diagonal sub-bands of second level DWT that contain

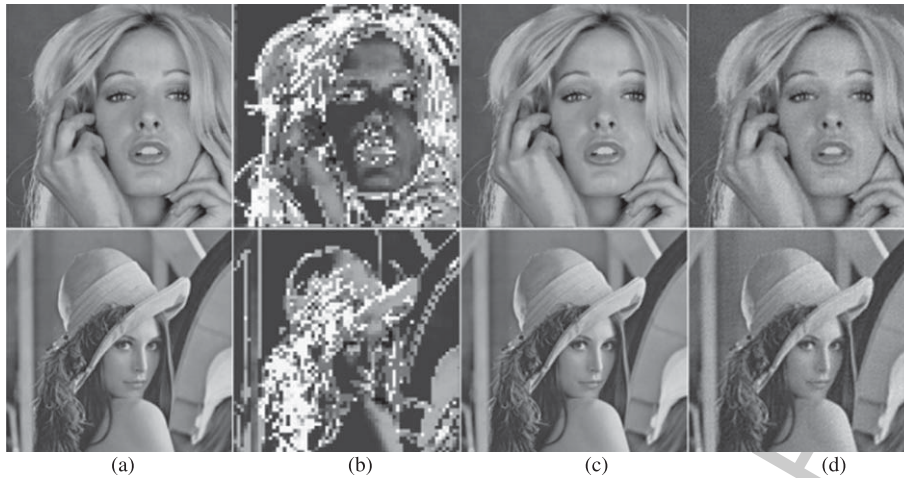


Fig. 9. Fuzzy strength-factor map. (a) The original image, (b) fuzzy map, (c) fuzzy watermarked image, (d) watermarked image using a simple method.

Table 1  
Summary of the proposed FIS fuzzy rules

Saliency	Edge-concentration	Intensity	Output
high	high	high	very strong
		medium	strong
		low	weak
	medium	high	strong
		medium	weak
		low	very weak
	low	high	moderate
		medium	weak
		low	very weak
		high	very strong
medium	high	high	medium
		medium	low
		low	not low
	medium	high	strong
		medium	weak
		low	not high
	low	high	strong
		medium	weak
		low	not high
		high	very strong
low	high	high	very strong
		medium	strong
		low	moderate
	medium	high	strong
		medium	strong
		low	weak
	low	high	strong
		medium	strong
		low	weak
		high	weak

LL	LLHL	HLHL	LHHL
LLLH	HLLH	LLHH	
LHLH			

Fig. 10. Selected sub-bands for embedding purpose.

### 3.3.1. Embedding algorithm

The general steps of the embedding procedure are shown in Fig. 11. For illustration purposes, we consider that the images are  $512 \times 512$  size and hence each designated sub-band, in Fig. 10, has the size of  $128 \times 128$ . However, the proposed method is general, and we can apply it to any size image.

Each sub-band is partitioned into  $8 \times 8$  blocks, and DCT of each block is computed. Hence, we have 256 DCT blocks in each sub-band. For improving watermark robustness, we embed the watermark with redundancy. The watermark length is 128 bits while each sub-band has an embedding capacity of 256 bits. Therefore, the watermark is embedded twice in each sub-band. Overall, this 128-bit string is embedded 14 times in the whole image. We can increase the length of the watermark string, but we have to reduce the embedding redundancy. For example, if the

high-frequency details are not considered because of their sensitivity to image processing attacks such as JPEG. Redundant watermark into multiple parts helps to create more robust embedding. Therefore, we redundantly embed in the seven sub-band levels, as shown in Fig. 10. In the following, the embedding and extraction methods are detailed.

Table 2  
NC results of our algorithm against cropping attacks

Image		Barbara		Baboon		Couple		Lena		Bridge	
		C	A	C	A	C	A	C	A	C	A
NC	5%	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99	1.00
	10%	1.00	0.91	1.00	0.95	1.00	0.95	0.91	1.00	0.91	1.00
	15%	1.00	0.91	1.00	0.95	1.00	0.94	0.88	1.00	0.87	1.00
	20%	1.00	0.91	1.00	0.95	1.00	0.93	0.88	1.00	0.88	1.00

C: Cropping center of the image. A: Cropping around the image.

Table 3  
Robustness of the proposed method against Gaussian attack using NC values

Image	$\sigma = 0.5$			$\sigma = 1$			$\sigma = 2$		
	$3 \times 3$	$5 \times 5$	$7 \times 7$	$3 \times 3$	$5 \times 5$	$7 \times 7$	$3 \times 3$	$5 \times 5$	$7 \times 7$
Lena	1.000	1.000	1.000	1.000	1.000	1.000	0.766	0.930	0.914
Barbara	1.000	1.000	1.000	0.969	0.992	0.992	0.820	0.875	0.898
Baboon	1.000	1.000	1.000	0.961	0.984	0.984	0.75	0.813	0.805
Bridge	1.000	1.000	1.000	0.961	0.938	0.953	0.695	0.773	0.711
Couple	1.000	1.000	1.000	0.984	0.992	0.984	0.820	0.875	0.914

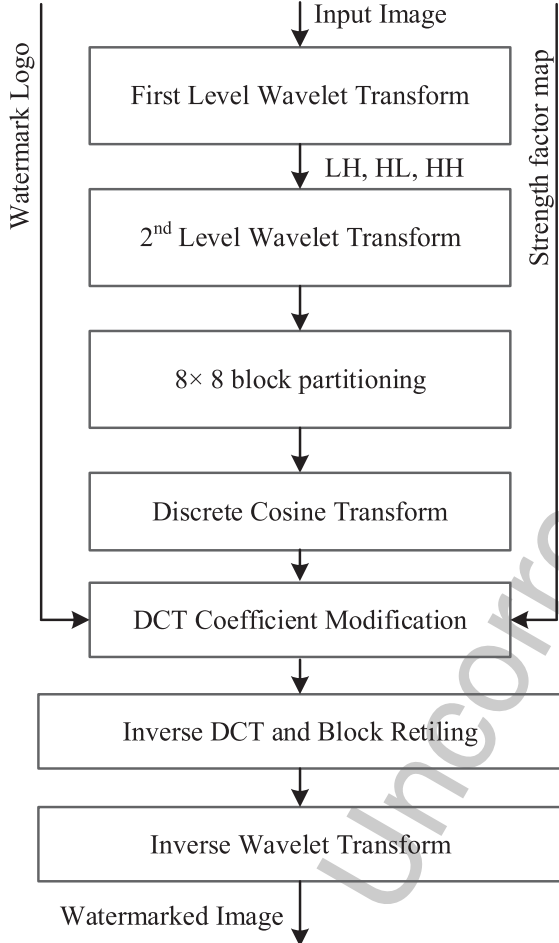


Fig. 11. Block diagram of the proposed embedding method.

watermark string length were to be 256, we only embed the string seven times in the whole image.

One bit of the watermark string is embedded into DCT coefficients of one  $8 \times 8$  block. The process to embed the  $j$ th copy of the  $i$ th watermark bit is performed in block  $B_{ij}$  based on (1).

$$\begin{cases} \text{for } W(B_{ij}) = 1, & \text{set } D(x, y) \geq D(u, v) + \alpha_{ij} \\ \text{for } W(B_{ij}) = 0, & \text{set } D(x, y) < D(u, v) + \alpha_{ij} \end{cases} \quad (1)$$

Variable  $W(B_{ij})$  shows the value of a binary watermark bit that is to be embedded in the block  $B_{ij}$ . Also,  $\alpha_{ij}$  is the adaptive strength factor for DCT block  $B_{ij}$ . This strength factor is computed by using the proposed fuzzy model. In an  $8 \times 8$  DCT block, two coefficients of  $D(u, v)$  and  $D(x, y)$  are used for the embedding process, where  $(u, v)$  and  $(x, y)$  are the coordinates of the two coefficients in that block. The relative values of these two coefficients show whether there is a 1 or 0 embedded in this block. If  $D(x, y)$  is greater than  $D(u, v)$ , by a margin of  $\alpha_{ij}$ , we consider that a watermark bit of 1 already exists in this block, otherwise we assume a 0 is present in the block. If, for example, a block by itself contains a 0 and we want to embed a 1 in this block, then the values of the coefficients are switched around, and their difference is enhanced by the strength factor, to satisfy the condition.

In the embedding process, we try to widen the difference between  $D(x, y)$  and  $D(u, v)$  using the strength factor such that the difference between these coefficients is maintained even after an attack.

Figure 12 shows the pseudo code of the proposed embedding algorithm. In this pseudo-code, we are

```

IF  $W(B_{ij}) = 1$  THEN
  IF  $D(x, y) \geq D(u, v)$  and  $|D(x, y) - D(u, v)| < \alpha_{ij}$ 
  THEN
     $D(x, y) \leftarrow D(x, y) + \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
     $D(u, v) \leftarrow D(u, v) - \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
  END
  IF  $D(x, y) < D(u, v)$  THEN
     $D(u, v) \leftarrow D(x, y) + \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
     $D(x, y) \leftarrow D(u, v) - \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
  END
ELSE
  IF  $D(x, y) < D(u, v)$  and  $|D(x, y) - D(u, v)| < \alpha_{ij}$ 
  THEN
     $D(x, y) \leftarrow D(x, y) + \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
     $D(u, v) \leftarrow D(u, v) - \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
  END
  IF  $D(x, y) \geq D(u, v)$  THEN
     $D(u, v) \leftarrow D(x, y) + \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
     $D(x, y) \leftarrow D(u, v) - \frac{|D(x, y) - D(u, v)| - \alpha_{ij}}{2}$ 
  END

```

Fig. 12. The pseudo-code of DCT coefficients modification for the proposed algorithm.

applying a strength factor of  $\alpha_{ij}$ . It means that if the values of the two coefficients of  $D(u, v)$  and  $D(x, y)$  are not different enough, the two coefficients are modified. The modified coefficients are guaranteed to be different enough based on the value of the strength factor  $\alpha_{ij}$ . This difference would help the survival of the watermark after signal processing attacks.

The mentioned steps of the pseudocode of Fig. 12 are repeated for all bits of the watermark string in each selected sub-band. After embedding a watermark bit in an  $8 \times 8$  block of a sub-band, we perform inverse DCT of that block. At the end, when all designated sub-bands are embedded in, inverse DWT is performed to generate the watermarked image. This process of redundant embedding increases the robustness and improves the extracted logo's visual fidelity in the presence of attacks.

### 3.3.2. Extraction scheme

Extraction is done completely blind which could be considered as one of the advantages of the proposed scheme. In blind methods, we do not need any additional information, such as the original image or the watermark string, for the extraction phase. Similar to embedding, the wavelet transform is done in horizontal, vertical and diagonal sub-band of the first level wavelet. A second level wavelet transform is applied to the HL, LH, and HH sub-bands to achieve a set of 12 sub-bands. Out of these 12 sub-bands, we select

Table 4  
Comparison of BER values for JPEG attack with different quality factors

Image	Quality factor				
	10	20	30	40	60
Lena	0.37	0.28	0.15	0.03	0.00
Barbara	0.29	0.19	0.08	0.01	0.00
Baboon	0.36	0.17	0.03	0.00	0.00
Bridge	0.22	0.07	0.03	0.02	0.00
Couple	0.29	0.15	0.04	0.00	0.00

Table 5  
Robustness of the proposed method and [37] using BER (%) values for Salt & Pepper attack

Salt & pepper	1%		3%		5%	
	ours	[37]	ours	[37]	ours	[37]
Barbara	0.04	3.28	<b>0.32</b>	4.18	<b>1.29</b>	4.80
Baboon	<b>0.00</b>	1.91	<b>0.38</b>	2.81	<b>0.71</b>	3.79
Couple	<b>0.15</b>	6.72	<b>0.62</b>	8.55	<b>5.28</b>	10.35
Bridge	<b>0.00</b>	5.43	<b>0.46</b>	7.27	<b>1.38</b>	9.10
Map	<b>0.00</b>	7.7	<b>0.00</b>	9.26	<b>0.26</b>	10.98

the seven sub-bands that are shown in Fig. 10. We then divide each of these seven sub-bands into  $8 \times 8$  blocks. After that, DCT transformation is applied to each block of the seven selected sub-bands. Extraction of the watermark string is performed based on (2):

$$W_x(B_{ij}) = \begin{cases} 1 & \text{if } D(x, y) \geq D(u, v) \\ 0 & \text{if } D(x, y) < D(u, v) \end{cases} \quad (2)$$

where  $W_x(B_{ij})$  is the extracted bit from block  $B_{ij}$ . Two 128-bit strings are extracted from each sub-band. Ultimately, fourteen ( $j = 1$  to 14) copies of the 128-bit ( $i = 1$  to 128) string are extracted from the watermarked image. We use (3) to perform voting and improve the robustness of the algorithm.

$$V_i = \sum_{j=1}^{14} W_x(B_{ij}) \quad (3)$$

where  $V_i$  shows the result of vote counting for the  $i$ th bit of the 128-bit string, where  $i = 1$  to 128. The voting is performed on all 14 versions of the extracted bit ( $j = 1$  to 14). The vote count,  $V_i$ , has a value between [0 14]. The voting result is based on (4) by assigning 0 or 1 for each bit of the final extracted string:

$$\bar{W}_i = \begin{cases} 1 & \text{if } V_i > 6 \\ 0 & \text{o.w.} \end{cases} \quad (4)$$

where  $\bar{W}_i$  is the final extracted watermark bit at position  $i$  and  $\bar{W}$  is the final extracted watermark string.



Table 6

BER (%) values for cropping attack. Comparing the proposed method with methods of [29, 40]

cropping	5%			10%		
	ours	[40]	[29]	ours	[40]	[29]
Lena	1.53	<b>0.78</b>	1.83	<b>4.69</b>	5.85	5.92
Goldhill	<b>0.00</b>	6.64	2.84	<b>4.69</b>	13.67	6.53
Bridge	<b>0.78</b>	8.98	4.63	<b>5.46</b>	18.36	13.42
Pepper	<b>0.00</b>	3.51	6.58	<b>4.69</b>	6.25	10.48

The embedding capacity of each sub-band is 256 bits. The watermark string determines the level of redundancy. If the length of the watermark is  $l$  the number of redundancy for each sub-band is  $256/l$  and the number of overall redundancy will be  $7 \times 256/l$ . Hence, we can use our method for different watermark lengths with different redundancies.

#### 4. Experimental results

In this Section, several experiments have been done to evaluate the performance of the proposed method. We use the common grayscale images that are used in [46]. All images are  $512 \times 512$ , and a randomly generated 128-bit string is used as the watermark. The Daubechies (db1) decomposition is performed for the wavelet transform. After taking DCT of each  $8 \times 8$  block, we select two of the coefficients. Experimentally we found out that coefficients at coordinates  $(u, v) = (5, 6)$  and  $(x, y) = (6, 5)$  have better robustness against most attacks. In addition, different coefficients of the fuzzy map are used for each of the seven-mentioned wavelet sub-bands. Experimentally, coefficient values of 0.45, 0.45 and 0.1 have been chosen for the fuzzy map generation of HL, LH and HH levels respectively. The LLHH fuzzy coefficient is more sensitive to changes. Hence, we embed with a higher strength factor as compared to the other sub-bands.

We compare the robustness and visual quality of our method with some other existing watermarking methods, such as [28, 29, 34, 37, 39, 40], which have same conditions as ours. To have a fair comparison, the same message length is considered in our experiments.

##### 4.1. Visual quality

Figure 13 shows results of applying our method to some standard images such as “Lena”, “Couple”, “Baboon” and “Lake”. Peak signal-to-noise ratio

(PSNR) is a simple and common metric for image quality. Due to ignoring the human visual system, PSNR is not an accurate quality measure. Thus, we use both PSNR and Mean structural similarity index (MSSIM) for estimating the perceptual quality of watermarked images. The proposed method keeps the high perceptual quality of the watermark image. MSSIM value for all of the watermarked images shown in Fig. 13 is 1.

##### 4.2. Robustness

To evaluate the robustness of our method, we calculate the normalized correlation ( $NC$ ) between original and extracted watermark after different attacks. These attacks contain salt and pepper noise (S&P), JPEG compression, cropping, Gaussian filter, median filter and white noise. The  $NC$  metric [47] was computed for each extracted string as follows:

$$NC = 1 - \frac{\sum_{i=1}^M W_i - \bar{W}_i}{M} \quad (5)$$

where  $W$  is the original and  $\bar{W}$  is the extracted watermark string and  $M$  is the string length. In some of the mentioned methods, the length of the string is different from ours. Hence in our comparisons, we consider this and if the string is smaller than ours, then the same string length is used but with higher redundancy.

To measure the robustness of our method, several attacks are applied to each image, and  $NC$  is computed for each string. Table 2 shows the  $NC$  results of our method against cropping attack. Different percentages of cropping (5%, 10%, 15%, 20%) are performed on mentioned image. Also, we considered two different cropping; the corner of the image and around the image. By increasing the percentage of cropping, our method has good robustness against attack in both kinds of cropping.

Another important attack is a Gaussian filter that we tested our method and evaluate its robustness. Three different window sizes  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$  with different standard deviations ( $\sigma = 0.5, 1, 2$ ) are used. The experimental results are indicated in Table 3. High  $NC$  values show the robustness of this method against Gaussian attacks. Once  $\sigma$  is chosen to be 0.5, our method has a good robustness regardless of the window size. By increasing the value of  $\sigma$ , the image is more smoothed and the frequency domain is more affected. Hence, the extracted watermark is

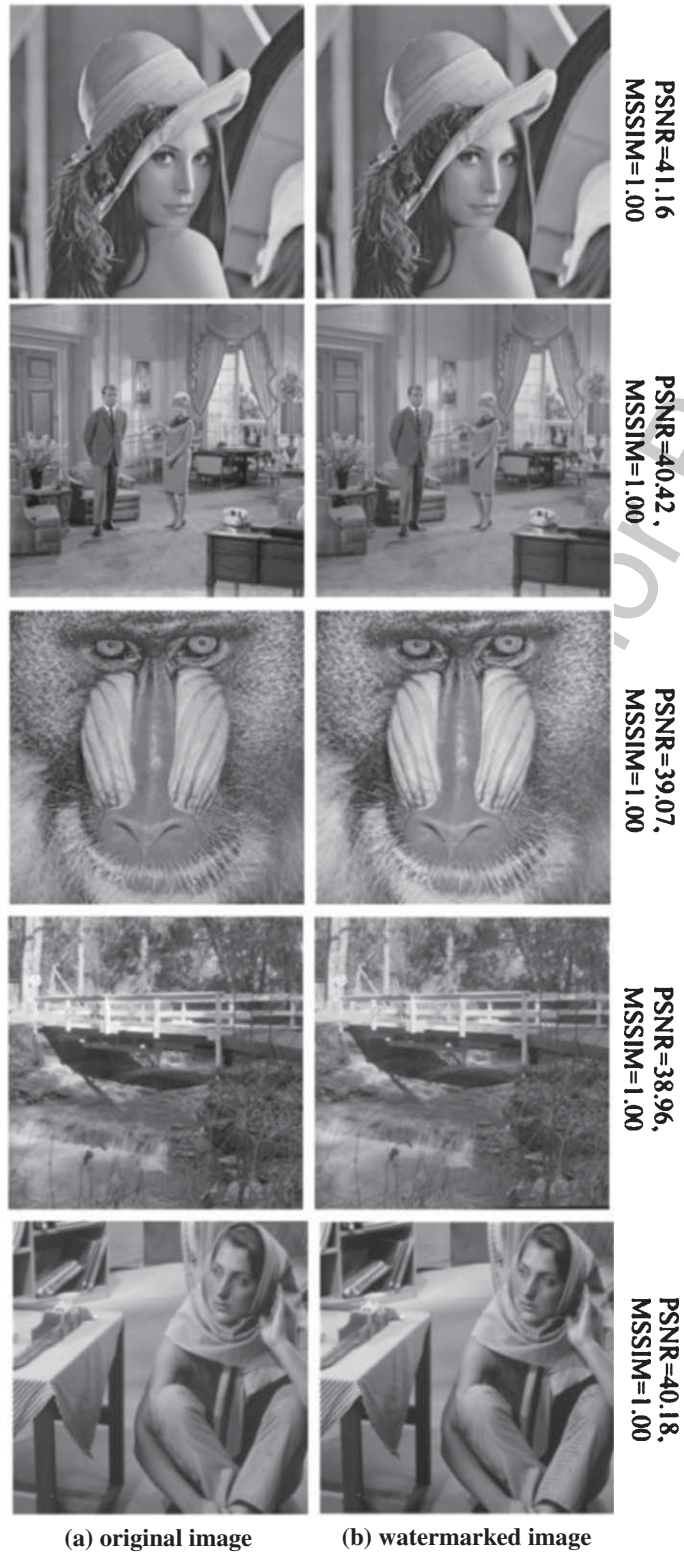


Fig. 13. The visual quality of the watermarked image as compared with the original image.



Table 7  
Robustness of the proposed method, [29, 39, 40] against the median filter and Gaussian attacks, using BER(%)

	Median filter $3 \times 3$				Gaussian filter $3 \times 3$			
	Ours	[40]	[39]	[29]	ours	[40]	[39]	[29]
Boat	14.8	<b>7.89</b>	16.42	–	<b>3.91</b>	4.69	9.39	–
Plane	17.1	<b>7.82</b>	11.8	–	6.25	<b>3.51</b>	6.8	–
Bridge	18.7	<b>11.72</b>	13.37	–	7.81	<b>5.86</b>	10.16	–
Pirate	12.5	<b>9.38</b>	11.34	–	12.5	5.86	<b>4.69</b>	–
Barbara	<b>11.71</b>	19.52	24.49	15.82	<b>3.91</b>	–	9.34	8.71
Baboon	<b>10.94</b>	17.38	33.63	15.14	10.16	–	17.58	<b>9.80</b>
Pepper	14.84	<b>8.30</b>	15.63	<b>4.41</b>	8.59	–	8.59	<b>3.55</b>
Goldhill	13.28	–	<b>1.65</b>	–	<b>6.25</b>	–	–	–
Lena	14.00	<b>6.24</b>	16.56	6.25	7.03	–	8.20	<b>4.73</b>

Table 8  
Robustness of the proposed method against white noise (AWGN) attack, compared to [28, 29, 40], using BER values, (length = 256 bit)

AWGN		Barbara	Baboon	Peppers	Lena
$\sigma_n$	10	Ours	1.56	<b>0.78</b>	3.13
		[28]	<b>1.40</b>	1.28	<b>1.32</b>
		[29]	4.48	1.09	1.87
		[40]	11.47	6.13	3.62
	20	Ours	<b>7.03</b>	<b>3.91</b>	<b>4.69</b>
		[28]	12.87	12.48	13.47
		[29]	–	–	–
		[40]	11.47	11.90	11.28
	30	Ours	<b>17.97</b>	<b>5.47</b>	<b>9.38</b>
		[28]	26.59	26.28	25.92
		[29]	–	–	–
		[40]	20.11	23.11	18.81

Table 9  
Algorithm complexity

Phase	CPU time (second)	Memory(MB)
Feature extraction	4.64	18.85
Fuzzy map generation	1.41	0.50
Embedding	0.28	12.00
Extraction	0.15	5.00

destroyed and the NC value is reduced. However, our method has acceptable robustness yet.

The robustness of our method against JPEG attacks is shown in Table 4. Image compression method of JPEG is one of the common attacks that our method has a good performance against it. In Table 4, some standard images, such as Barbara, Lena, Baboon, Bridge, and Couple, are used, and bit error rate (BER) percentage values are computed. Low BER indicated the robustness of a method against these attacks. When the JPEG quality is 60 or more, we can extract the watermark with no error.

#### 4.3. Comparison with state of the art methods

To evaluate the performance of the proposed method, we compare our method with recent algorithms of [28, 29, 34, 37, 39, 40]. These methods

embed in DWT and CT respectively. All of these methods have used the same watermark length as we do. In each paper, some images with special attacks are considered. Hence, we compare our method with them based on their reported results. Bit error rates (BER) percentages are shown in Tables 5 to 10. Lena, Barbara, Boat, Baboon, Couple, Bridge, Goldhill, Pepper, Pirate, and Jetplane are some standard images that are used in this comparison. Different attacks, such as salt & pepper (1%, 3%, 5%), cropping (5%, 10%, 20%), Median filter ( $3 \times 3$ ), Gaussian filter ( $3 \times 3$ ), white noise, are considered here. These results reveal that our method is highly robust against different mentioned attacks. Adaptive fuzzy strength factor helps us to have good robustness while it keeps a reasonable degree of imperceptibility.

Tables 5–10 show comparisons between our method and methods of [28, 29, 34, 37, 39, 40]. In all of these comparisons, we try to produce images that have PSNR values similar to the mentioned references.

For attacks such as salt & pepper, Gaussian noise, white noise, we repeat the experiments for 10 times, and then the average of all BER values is indicated.

Table 5 shows BER values of the method of [37] and ours against salt & pepper attack. Different percentages of salt and pepper noise (1%, 3%, 5%) are examined. As we can see, in all cases our method performs better than the method of [37]. Based on Table 6, our method has better results for ten percent cropping attack. When 5% cropping is applied, our method performs well for three out of the four images.

In Table 7, the results of [29, 39, 40] are shown against Median and Gaussian filters. As we can see, our method does not have good results in median filter attack. However, our method has comparable results against Gaussian filter attack in comparison to other methods. These accuracy numbers are extracted from [40].

Table 10  
Robustness of the proposed method against different attacks compared to [34] using BER values

Image	Method	MSSIM		JPEG Attack								S&P 1%	Cropping 20%	Sharpening	Rotation 0.5'	Median filter
			10%	20%	30%	40%	50%	60%	70%	80%	90%					
Bridge	ours	<b>1.000</b>	<b>0.227</b>	<b>0.078</b>	<b>0.031</b>	<b>0.008</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.531	<b>0.203</b>
	[34]	0.9966	0.418	0.383	0.363	0.344	0.356	0.359	0.359	0.352	0.352	0.355	0.389	0.355	<b>0.465</b>	0.379
Boat	ours	<b>1.000</b>	<b>0.328</b>	<b>0.125</b>	<b>0.078</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.055</b>	<b>0.000</b>	0.562	<b>0.148</b>
	[34]	0.9969	0.441	0.387	0.391	0.387	0.379	0.367	0.356	0.352	0.348	0.367	0.391	0.354	<b>0.488</b>	0.367
Barbara	ours	<b>1.000</b>	<b>0.328</b>	<b>0.188</b>	<b>0.070</b>	<b>0.008</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.609	<b>0.125</b>
	[34]	0.9970	0.426	0.406	0.379	0.371	0.375	0.363	0.359	0.352	0.348	0.390	0.393	0.359	<b>0.490</b>	0.367
Jet plane	ours	<b>1.000</b>	0.438	<b>0.258</b>	<b>0.078</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.070</b>	<b>0.000</b>	0.586	<b>0.179</b>
	[34]	0.9976	<b>0.434</b>	0.402	0.391	0.391	0.387	0.371	0.359	0.355	0.348	0.387	0.412	0.357	<b>0.455</b>	0.379
Lena	ours	<b>1.000</b>	<b>0.352</b>	<b>0.297</b>	<b>0.140</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.602	<b>0.132</b>
	[34]	0.9980	0.469	0.418	0.398	0.379	0.379	0.383	0.359	0.355	0.348	0.395	0.414	0.365	<b>0.475</b>	0.383
Lake	ours	<b>1.000</b>	<b>0.336</b>	<b>0.156</b>	<b>0.046</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.570	<b>0.195</b>
	[34]	0.9971	0.465	0.410	0.379	0.375	0.367	0.363	0.359	0.363	0.348	0.387	0.402	0.355	<b>0.492</b>	0.363
Couple	ours	<b>1.000</b>	<b>0.328</b>	<b>0.148</b>	<b>0.039</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.523	<b>0.109</b>
	[34]	0.9976	0.414	0.398	0.395	0.387	0.363	0.359	0.356	0.359	0.352	0.371	0.387	0.348	<b>0.475</b>	0.387
Baboon	ours	<b>1.000</b>	<b>0.188</b>	<b>0.039</b>	<b>0.007</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.055</b>	<b>0.000</b>	0.477	<b>0.094</b>
	[34]	0.9958	0.398	0.375	0.356	0.363	0.352	0.348	0.352	0.359	0.355	0.355	0.389	0.357	<b>0.510</b>	0.363
Map	ours	<b>1.000</b>	<b>0.094</b>	<b>0.047</b>	<b>0.015</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.070</b>	<b>0.000</b>	0.562	<b>0.250</b>
	[34]	0.9948	0.379	0.356	0.356	0.363	0.344	0.348	0.344	0.352	0.348	0.348	0.393	0.359	<b>0.471</b>	0.359
Peppers	ours	<b>1.000</b>	<b>0.414</b>	<b>0.227</b>	<b>0.023</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.617	<b>0.164</b>
	[34]	0.9976	0.441	0.398	0.391	0.375	0.371	0.363	0.352	0.348	0.348	0.406	0.395	0.355	<b>0.482</b>	0.383
Pirate	Ours	<b>1.000</b>	<b>0.265</b>	<b>0.063</b>	<b>0.023</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.609	<b>0.125</b>
	[34]	0.9970	0.453	0.418	0.371	0.383	0.367	0.363	0.359	0.355	0.355	0.391	0.396	0.359	<b>0.484</b>	0.387
average	Ours	<b>1.000</b>	<b>0.299</b>	<b>0.148</b>	<b>0.050</b>	<b>0.001</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.000</b>	<b>0.063</b>	<b>0.000</b>	0.568	<b>0.157</b>
	[34]	0.9969	0.431	0.431	0.396	0.379	0.374	0.367	0.362	0.356	0.350	0.377	0.396	0.357	<b>0.481</b>	0.374

In Table 8, we compare our method against additive white Gaussian noise attack, and our method shows better results. The watermark string length is 256 bits.

We also compare our method with [34] that is a recent watermarking algorithm. Different attacks, such as JPEG, median filter, salt and pepper (1 %), cropping, sharpening, and rotation, are considered. As shown in Table 10, our fuzzy method has better results for different attacks except for the median filter. Also, our method has higher MSSIM results than [34].

For further evaluation of the performance of the proposed method, we tested our algorithm on the CorelDraw dataset. This dataset contains 8185 grayscale images with the size of  $512 \times 512$ . We randomly selected 100 images and compared the average performance of the proposed method with the average values we are reporting in Table 10. We also applied different attacks on the mentioned 100 watermarked images. The results of the proposed method applied to the CorelDraw images are very close to those obtained from the standard images. This shows that our proposed fuzzy method, overall, has good robustness and visual quality for different images.

#### 4.4. Algorithm performance

The proposed method is implemented on a computer with Intel® Core™ i5-2410M CPU @ 2.30GHz and a 64-bit operating system. Also, MATLAB R2016a is used for implementation of our simulations. Here we report the complexity in terms of memory usage and CPU time to show the performance of the proposed algorithm.

In Table 9, we show the complexity of different phases of the proposed method. As can be seen, feature extraction is the most time-consuming phase, partially due to the complexity of the saliency detection phase of the algorithm. Parallel implementation of different phases of the algorithm could be used to decrease the complexity of the proposed method. Since the proposed method is blind, feature extraction is not performed in the extraction phase. Therefore, the memory usage in the decoder is less than the encoder.

## 5. Conclusion

In this paper, we proposed a novel adaptive blind watermarking using the fuzzy system. The proposed

algorithm tries to use human psycho-visual characteristics to choose appropriate locations for embedding the watermark bits. At first, a fuzzy inference system is fed with three attributes: saliency, intensity, and edge-concentration of the original image. The FIS produces a map that is used to determine adaptive embedding strength factor of every image block. The fuzzy system results in severe embedding in regions that are not salient, contain edges, and have high intensities. After producing a fuzzy map, the wavelet transform is applied to the original image. Some of the second level sub-bands are used for embedding. Block level DCT of these sub-bands are computed. Modification of some of the DCT coefficients would perform the final embedding task. The combination of DWT and DCT resulted in the high visual quality of the watermarked images. We tested our method against different attacks, and experimental results show that this method has better performance than comparable state-of-the-art methods.

## References

- [1] A. Rashid, Digital watermarking applications and techniques: A brief review, *Int Journal of Computer Applications Technology and Research* 5(3) (2016), 147–150.
- [2] J. Liu and X. He, A review study on digital watermarking, in *Proc IEEE International Conference on Information and Communication Technology*, 2005, pp. 337–341.
- [3] V.M. Potdar, S. Han and E. Chang, A survey of digital image watermarking techniques, in *Proc IEEE International Conference on Industrial Informatics*, 2005, pp. 709–716.
- [4] S. Roy and Q. Sun, Robust hash for detecting and localizing image tampering, in *Proc IEEE Int Conf Image Process (ICIP)*, vol. 6, 2007, pp. 117–120.
- [5] M. Tagliasacchi, G. Valenzise and S. Tubaro, Hash-based identification of sparse image tampering, *IEEE Trans Image Process* 18(11) (2009), 2491–2504.
- [6] W.H. Lin, S.J. Horng, T.W. Kao, R.J. Chen, Y.H. Chen, C.L. Lee and T. Terano, Image copyright protection with forward error correction, *Expert Systems with Applications* (2009), 11888–11894.
- [7] T. Zong, Y. Xiang, I. Natgunanathan, S. Guo, W. Zhou and G. Beliaikov, Robust histogram shape-based method for image watermarking, *IEEE Transactions on Circuits and Systems for Video Technology* (2014), 717–729.
- [8] H. Zhang, H. Shu, G. Coatrieux, J. Zhu, Q.J. Wu, Y. Zhang, H. Zhu and L. Luo, Affine legendre moment invariants for image watermarking robust to geometric distortions, *IEEE Transactions on Image Processing* (2011), 2189–2199.
- [9] V. Solachidis and P. Ioannis, Circularly symmetric watermark embedding in 2-D DFT domain, *IEEE Transactions on Image Processing* (2001), 1741–1753.
- [10] M.A. Suhail and M.S. Obaidat, Digital watermarking-based DCT and JPEG model, *IEEE Transactions on Instrumentation and Measurement* (2003), 1640–1647.

- [11] A.H. Taherinia and M. Jamzad, A robust spread spectrum watermarking method using two levels DCT, *International Journal of Electronic Security and Digital Forensics* (2009), 280–305.
- [12] S.A. Parah, J.A. Sheikh, N.A. Loan and G.M. Bhat, Robust and blind watermarking technique in DCT domain using inter-block coefficient differencing, *Digital Signal Processing*, vol. 53, 2016, pp. 11–24.
- [13] M. Heidari, S. Samavi, S.M.R. Soroushmehr, S. Shirani, N. Karimi and K. Najarian, Framework for robust blind image watermarking based on classification of attacks, *Multimedia Tools and Applications* (2016), 1–21.
- [14] F. Huang and Z.H. Guan, A hybrid SVD-DCT watermarking method based on LPSNR, *Pattern Recognition Letters* (2004), 1769–1775.
- [15] S.J. Horng, D. Rosiyadi, P. Fan, X. Wang and M.K. Khan, An adaptive watermarking scheme for e-government document images, *Multimedia Tools and Applications* (2014), 3085–3103.
- [16] S. Oueslati, A. Cherif and B. Solaiman, A fuzzy watermarking approach based on the human visual system, *International Journal of Image Processing* (2010), 218–231.
- [17] B. Jagadeesh, P. Rajesh Kumar and P. Chenna Reddy, Robust digital image watermarking based on fuzzy inference system and back propagation neural networks using DCT, *Soft Computing* (2016), 3679–3686.
- [18] B. Jagadeesh, P.R. Kumar and P.C. Reddy, Fuzzy inference system based robust digital image watermarking technique using discrete cosine transform, *Procedia Computer Science* (2015), 1618–1625.
- [19] P. Meerwald and U. Andreas, Survey of wavelet-domain watermarking algorithms, *International Society for Optics and Photonics* (2001), 505–516.
- [20] V.M. Potdar, S. Han and E. Chang, A survey of digital image watermarking techniques, in *Proc IEEE International Conference on Industrial Informatics*, 2005, pp. 709–716.
- [21] P. Rasti, G. Anbarjafari and H. Demirel, Colour image watermarking based on wavelet and QR decomposition, in *Proc of IEEE Signal Processing and Communications Applications Conference*, 2017, pp. 1–4.
- [22] C.C. Lai and C.C. Tsai, Digital image watermarking using discrete wavelet transform and singular value decomposition, *IEEE Transactions on Instrumentation and Measurement* (2010), 3060–3063.
- [23] W.H. Lin, Y.R. Wang and S.J. Horng, A wavelet-tree-based watermarking method using distance vector of binary cluster, *Expert Systems with Applications* (2009), 9869–9878.
- [24] W.H. Lin, Y.R. Wang, S.J. Horng, T.W. Kao and Y. Pan, A blind watermarking method using maximum wavelet coefficient quantization, *Expert Systems with Applications* (2009), 11509–11516.
- [25] W.H. Bin, Y.H. Liang, W.C. Dong and W.S. Ming, A New Watermarking Algorithm Based on DCT and DWT Fusion, in *Proc International Conference on Electrical and Control Engineering*, 2010, pp. 2614–2617.
- [26] A. Al-Haj, Combined DWT-DCT digital image watermarking, *Journal of Computer Science* (2007), 740–746.
- [27] E. Ganic and A.M. Eskicioglu, Robust DWT-SVD domain image watermarking: Embedding data in all frequencies, in *Proc workshop on Multimedia and Security*, 2004, 166–174.
- [28] E. Nezhadarya, Z.J. Wang and R.K. Ward, Robust image watermarking based on multiscale gradient direction quantization, *IEEE Transactions on Information Forensics and Security* 6(4) (2011), 1200–1213.
- [29] M. Hamghalam, S. Mirzakuchaki and M.A. Akhaee, Geometric modeling of the wavelet coefficients for image watermarking using optimum detector, *IET Image Processing* 8(3) (2014), 162–172.
- [30] M.S. Hsieh, Perceptual copyright protection using multiresolution wavelet-based watermarking and fuzzy logic, arXiv preprint arXiv:1007.5136, 2010.
- [31] M.C. Motwani, R.C. Motwani and F.C. Harris, Wavelet based fuzzy perceptual mask for images, in *Proc IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 4261–4264.
- [32] V.S. Jabade and S.R. Gengaje, Logo based image copyright protection using discrete wavelet transform and fuzzy inference system, *International Journal of Computer Applications* (2012).
- [33] S. Ghannam and F.E. Abou-Chadi, Enhancing robustness of digital image watermarks using Contourlet transform, in *Proc IEEE International Conference on Image Processing*, 2009, pp. 3645–3648.
- [34] H.R. Fazlali, S. Samavi, N. Karimi and S. Shirani, Adaptive blind image watermarking using edge pixel concentration, *Multimedia Tools and Applications*, 2017, pp. 3105–3120.
- [35] E. Etemad, S. Samavi, S.R. Soroushmehr, N. Karimi, M. Etemad, S. Shirani and K. Najarian, Robust image watermarking scheme using bit-plane of Hadamard coefficients, *Multimedia Tools and Applications* (2017), 1–23.
- [36] H.R. Kaviani, S. Samavi, N. Karimi and S. Shirani, Elevating watermark robustness by data diffusion in Contourlet coefficients, *IEEE International Conference on Communications (ICC)*, 2012, pp. 6739–6743.
- [37] A. Akhaee, S.M. Sahraeian and F. Marvasti, Contourlet-based image watermarking using optimum detector in noisy environment, *IEEE Transactions on Image Processing* 19 (2010), 967–980.
- [38] H. Sadreazami, M.O. Ahmad and M.N. Swamy, Multiplicative watermark decoder in contourlet domain using the normal inverse gaussian distribution, *IEEE Transactions on Multimedia* (2016), 196–207.
- [39] M. Akhaee, S. Sahraeian and C. Jin, Blind image watermarking using a sample projection approach, *IEEE Transactions on Information Forensics and Security* 6(3) (2011), 883–893.
- [40] M. Hamghalam, S. Mirzakuchaki and M.A. Akhaee, Robust image watermarking using dihedral angle based maximum-likelihood detector, *IET Image Processing* 7(5) (2013), 451–463.
- [41] J. Harel, C. Koch and P. Perona, Graph-based visual saliency, in *NIPS*, 2006.
- [42] O. Castillo, M.A. Sanchez, C.I. Gonzalez and G.E. Martinez, Review of recent type-2 fuzzy image processing applications, *Information* 8(3) (2017), 97.
- [43] C.I. Gonzalez, P. Melin, J.R. Castro, O. Mendoza and O. Castillo, An improved sobel edge detection method based on generalized type-2 fuzzy logic, *Soft Computing* 20(2) (2016), 773–784.
- [44] E. Pourjavad and R.V. Mayorga, A comparative study and measuring performance of manufacturing systems with Mamdani fuzzy inference system, *Journal of Intelligent Manufacturing* (2017), 1–13.
- [45] Y. Zheng, B. Jeon, D. Xu, Q.M. Wu and H. Zhang, Image segmentation by generalized hierarchical fuzzy C-means algorithm, *Journal of Intelligent & Fuzzy Systems* 28(2) (2015), 961–73.
- [46] Availabelat: <http://decsai.ugr.es/cvg/dbimagenes/index.php>.

- 1047 [47] C.S. Shieh, H.C. Huang, F.H. Wang and J.S. Pan, Genetic watermarking based on transform-domain techniques, *Pat-*  
1048 *tern Recognition* (2004), 555–565. 1052  
1049  
1050 [48] H.C. Nothdurft, Saliency from feature contrast: Additivity 1053  
1051 across dimensions, *Vision Research* (2000), 1183–1201. 1054  
1055  
1056 [49] R. Carmi and L. Itti, Visual causes versus correlates of atten-  
tional selection in dynamic scenes, *Vision Research* (2006),  
4333–4345. 1057  
[50] R.C. Gonzalez and R.E. Woods, Digital image processing,  
2012. 1058

Uncorrected Author Proof