

## DFT-Based Reversible Watermarking Method for Image Ownership Protection

Ansam Osamah Abdulmajeed\* and Sundus Abdulmuttalib Mohamed

College of Computer Science and Mathematics, University of Mosul, Almajmoa, 41002, Mosul, Iraq

### ABSTRACT

The Performance of Discrete Fourier Transform (DFT)-based watermarking methods has been carefully examined in the literature. Although the watermark in most of the literature was embedded in the DFT magnitudes using bit plane embedding, it was recently embedded in the Direct Current (DC) coefficient in the spatial domain. However, data loss due to rounding and replacement operations are still evident. Therefore, the objective of the method proposed here was to combine previous literature designs to implement a reversible DFT-based watermarking method for image ownership protection using bit plane embedding in the DC coefficient. The watermark was embedded in a middle bit plane of the DC coefficient for each DFT-transformed image block. In order to ensure reversibility and improve the security level, a combination of double Feynman and XOR gates was used to shuffle the watermark bits. The results revealed that the 8<sup>th</sup> (PSNR/SSIM = 32dB/0.8826), the 9<sup>th</sup> (PSNR/SSIM = 38dB/0.0.9587), and the 10<sup>th</sup> (PSNR/SSIM = 44dB/0.9917) bit planes for block sizes of 4×4, 8×8, and 16×16, respectively, were the best bit planes showing good imperceptibility and resistance to compression, filtering, and noise attacks. In conclusion, embedding the DC coefficients rather than all the magnitudes has influentially increased the watermarking robustness. In contrast, embedding the DC coefficients in the frequency domain rather than the spatial domain reduced the image's structural contents distortion.

Furthermore, the proposed method for grayscale images is effective in applications where reversibility is desired. However, further studies to find colored images' reversible methods are recommended.

**Keywords:** Bit plane, DC coefficient, digital watermarking, discrete Fourier transform, Feynman gate, reversible watermarking

### ARTICLE INFO

#### Article history:

Received: 6 November 2021

Accepted: 31 January 2022

Published: 20 April 2022

DOI: <https://doi.org/10.47836/pjst.30.3.07>

#### E-mail addresses:

ansam\_osamah@uomosul.edu.iq (Ansam Osamah Abdulmajeed)

sundus\_abid7@uomosul.edu.iq (Sundus Abdulmuttalib Mohamed)

\*Corresponding author

## INTRODUCTION

Protection of intellectual rights has recently become crucial due to the huge amounts of digital media being increasingly transmitted throughout the internet. Watermarking is one of the most powerful techniques to protect digital media from intentional or unintentional tampering behaviors (Feng et al., 2019). It is widely used in applications of ownership pretensions and confirming copyrights (Luo et al., 2021). Digital watermarking embeds watermark information in the host media in some way that prevents it from being destroyed and being matched later with those of the owners (Feng et al., 2019). The basic requirements that must be taken into consideration in designing a digital watermarking system are imperceptibility, payload, and robustness, where imperceptibility means the amount of perceptual effect of an embedded watermark on the quality of the host media, while payload refers to the amount of information that can be embedded in the host media without affecting the quality (Qasim et al., 2018). Robustness denotes the ability of a watermarking method to resist attacks. However, not all watermarking methods can resist all types of attacks. For instance, fragile watermarking methods cannot resist slight tampering of host media. On the other hand, robust watermarking methods can resist intentional and malicious attacks. There are semi-fragile methods that can withstand only unintended attacks, such as noise attacks and JPEG compression (Menendez-Ortiz et al., 2019). There is a trade-off between payload and both robustness and imperceptibility. A large payload can be acquired by compromising either robustness or imperceptibility (Ansari et al., 2018).

In some critical applications, such as medical applications, reversible watermarking methods are required to restore the original copy of the host media. Indeed, reversible watermarking methods recover the host media without any loss if the channel is noise-free. In contrast, a complete recovery of the host media cannot be guaranteed if there are attacks (Menendez-Ortiz et al., 2019).

Watermarking methods can be divided into three categories according to the extraction process. Blind extraction does not require additional information other than the secret key to retrieve the watermark. Non-blind extraction, on the other hand, necessitates both the watermark and the original media, whereas semi-blind extraction necessitates the watermark to extract the embedded watermarks (Khalilidan et al., 2020). A watermark is embedded either in the spatial or frequency domain. In the frequency domain, the most commonly used transforms are the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT) (Jimson & Hemachandran, 2018). The performance of DFT-based digital watermarking methods has been carefully examined in various studies (Table 1). For instance, Ahmed and Moskowitz (2004) proposed a bit plane embedding method to embed the watermark in the middle bit plane of the rounded magnitude. This study found that the 13<sup>th</sup> bit plane was the optimum plane for embedding. In additional work, Ahmed and Moskowitz (2006) proposed a semi-reversible method where the bit plane

(i-1) was replaced with the embedding bit plane (i) prior to the embedding process to recover the most impact bit plane on image quality during the extraction process.

In another study, Zhu et al. (2007) examined the fragility of the method proposed by Ahmed and Moskowitz (2004) against malicious attacks. This study found that the embedding in the 9<sup>th</sup> bit plane did not resist any unintended or malicious attacks but reserved good image quality. In contrast, embedding in the 13<sup>th</sup> bit plane showed good resistance to JPEG compression; however, it was very fragile against tampering. On the other hand, Su et al. (2019) embedded the watermarking bits by modifying the pixels of colored image blocks based on the Direct Current (DC) coefficients, which were calculated in the spatial domain, and their optimal boundary values. Finally, Zhang et al. (2020) proposed a blind color image watermarking algorithm in the spatial domain. The embedding strategy in this study was designed based on the similarity between the values of the DC coefficients of adjacent blocks.

Table 1

*The design of the related DFT-based watermarking method*

Method	Domain	Host image	Watermark image	Embedding technique	Place of embedding	Reversibility
Ahmed and Moskowitz (2004)	Frequency	512×512 grayscale	512×512 Binary	Bit plane embedding	Real magnitudes	Irreversible
Ahmed and Moskowitz (2006)	Frequency	512×512 grayscale	512×512 Binary	Bit plane embedding	Real magnitudes	Semi-reversible
Zhu et al. (2007)	Frequency	512×512 grayscale	512×512 Binary	Bit plane embedding	Real magnitudes	Irreversible
Su et al. (2019)	Spatial	512×512 RGB	32×32 RGB	Quantization technique	DC coefficient	Irreversible
Zhang et al. (2020)	Spatial	512×512 RGB	32×32 RGB	Quantization technique	DC coefficient	Irreversible

Previous studies, however, suffered from data loss due to rounding and replacement operations which made them irreversible. Therefore, the objective of the proposed method here was to implement a reversible watermarking method. The current method combines the designs of the abovementioned studies listed in Table 1. It implemented a bit plane embedding in the DC coefficient in the frequency domain. The watermark was embedded in a middle bit plane of the DC coefficient for each DFT-transformed image block (Figure 1). In order to ensure reversibility and improve the security level, the proposed method used a combination of double Feynman and XOR gates to shuffle the watermark bits with both the secret key and the original bit of the selected bit plane.

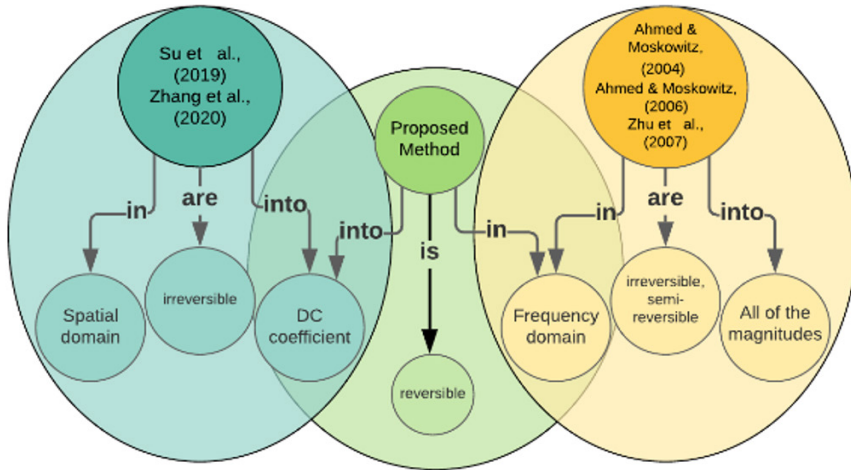


Figure 1. Proposed method framework

## MATERIALS AND METHOD

### Discrete Fourier Transform (DFT)

DFT is one of the most widely used transforms in the digital watermarking algorithm. For example, the digital image can be transformed into the DFT domain by Equation 1 below (Qasim et al., 2018):

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j2\pi(\frac{ux}{N} + \frac{vy}{M})} \quad [1]$$

where  $M$  and  $N$  are the image dimensions, and  $f(x, y)$  is the image's pixel. DFT coefficients,  $F(u, v)$ , are complex numbers. In polar form, the real  $R(u, v)$  and imaginary  $I(u, v)$  parts of the coefficients are expressed as the magnitude and phase, which can be computed by Equations 2 and 3, respectively (Qasim et al., 2018):

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}} \quad [2]$$

$$\angle(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right] \quad [3]$$

The phase is more significant than the magnitude since the magnitude includes fewer pieces of information (Jimson & Hemachandran, 2018).  $F(0, 0)$  is often called the DC coefficient, representing the image's average brightness (Zhang et al., 2020). The Inverse Discrete Fourier Transform (IDFT) can be computed by the following Equation 4 (Qasim et al., 2018):

$$f(x, y) = \frac{1}{NM} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{j2\pi(\frac{ux}{N} + \frac{vy}{M})} \quad [4]$$

### Feynman Gate

The reversible logical gate maps one input to one output without data loss, e.g., NOT and XOR gates. Feynman gate is a reversible logical gate of two inputs and two outputs. The first input is the control and the second one is the target. The Feynman gate negates the target input only if the control input is set. Therefore, it is known as the ‘‘Controlled NOT’’ gate. The Double Feynman gate comprises two Feynman gates (Figure 2) (Krishna & Ramesh, 2019).

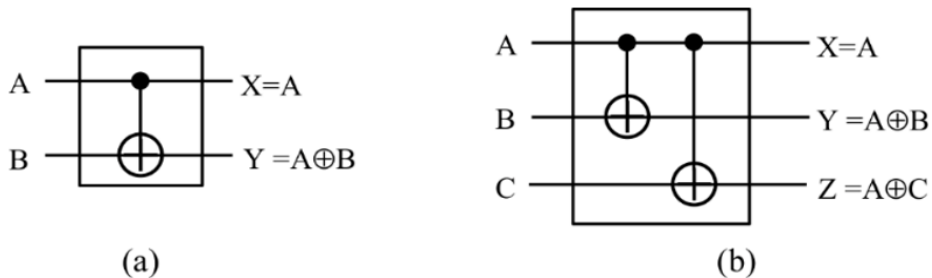


Figure 2. (a) Feynman gate (b) Double Feynman gate (Krishna & Ramesh, 2019)

### Proposed Method

**Embedding Process.** The in-detail steps of the embedding process included the following (Figure 3):

**Step-1:** The host image  $I$ , of size  $M \times M$ , was subdivided into non-overlapping blocks of size  $N \times N$ .

**Step-2:** Each block was transformed to the frequency domain using DFT.

**Step-3:** The original watermark  $W$  was shuffled with the secret key  $K$  and the selected middle bit plane  $BP$  of the DC coefficient of each block using a combination of double Feynman and XOR gates. Both  $W$  and  $K$  were binary images of size  $M/N \times M/N$ .

**Step-4:** Every bit of the shuffled watermark  $W_s$  was embedded in the selected middle bit plane  $BP$  of the DC coefficients.

**Step-5:** IDFT was applied to produce the watermarked image  $I_w$ .

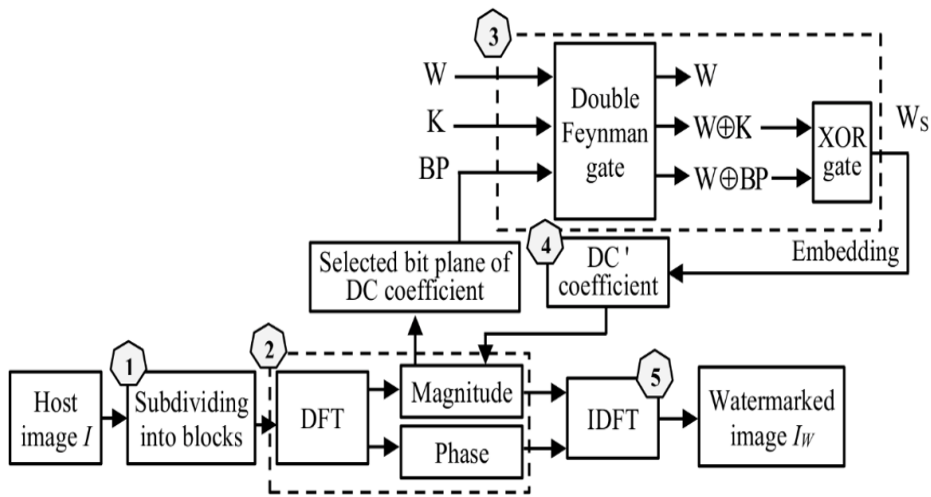


Figure 3. The embedding process of the proposed method

**Extraction Process.** The extraction process required the original watermark to extract the hidden watermark because the proposed method was a semi-blind watermarking method. The extraction process included the following steps (Figure 4):

**Step-1:** The watermarked image  $I_W$  was subdivided into non-overlapping blocks of size  $N \times N$ .

**Step-2:** Each block was transformed to the frequency domain using DFT.

**Step-3:** The hidden watermark  $W_S$  was extracted from the selected bit plane of the DC coefficients of each block.

**Step-4:** The original watermark  $W$  was re-initialized using the secret key  $K$ , such that  $W_{RE} = W \oplus K$ .

**Step-5:**  $W_S$  was passed to XOR gate along with  $W_{RE}$ .

**Step-6:** The result of Step 5 was passed to XOR twice with the following:

**1st.** The original watermark  $W$  to recover the original bit plane of the DC coefficient, replaced by those of each block.

**2nd.** The recovered bit plane of the DC coefficient to extract the watermark.

**Step-7:** The magnitude and the unchanged phase were passed to IDFT to recover an identical copy of the host image.

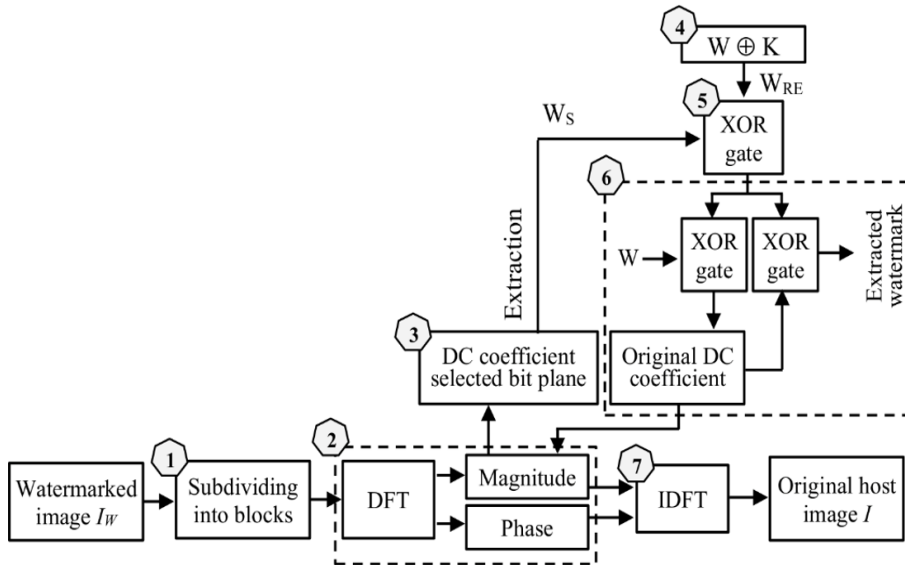


Figure 4. The extraction process of the proposed method

**Analysis of the Performance**

Several metrics can ensure that the digital watermarking methods achieve the most important requirements of secure watermarking. Peak Signal-to-Noise Ratio (PSNR) is the most widely measure used to test the method’s imperceptibility. PSNR can be computed according to the following Equation 5 (Qasim et al., 2018):

$$PSNR(I, I_w) = 10 \times \log_{10} \frac{MAX_I^2}{MSE} \tag{5}$$

where  $I$  and  $I_w$  are the host and watermarked images, respectively.  $MAX$  denotes the value of the maximum sample, and  $MSE$ , the Mean Square Error (Equation 6) (Qasim et al., 2018):

$$MSE(I, I_w) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - I_w(i, j))^2 \tag{6}$$

The main component of PSNR is the squared error values that refer to the difference between the co-located pixel values. The accepted value of PSNR is greater than 30 dB (Setiadi, 2021). When the two images are identical, the value of the corresponding PSNR is infinite. Accordingly, the value of PSNR must be sufficiently large to assert that the watermarking methods are imperceptible (Begum & Uddin, 2021). Structural Similarity Index (SSIM) is a more recent measurement than PSNR that measures the similarity between two images based on the luminance, contrast, and correlation coefficients (Setiadi,

2021). The SSIM between the host image  $I$  and the watermarked image  $I_w$  can be computed as the following Equation 7 (Qasim et al., 2018):

$$SSIM(I, I_w) = \frac{(2\mu_I\mu_{I_w} + c_1)(2cov + c_2)}{(\mu_I^2 + \mu_{I_w}^2 + c_1)(\sigma_I^2 + \sigma_{I_w}^2 + c_2)} \quad [7]$$

where  $\mu$ ,  $\sigma$ , and  $cov$  are the mean, standard deviations, and covariance, which measure the luminance, the contrast, and the structure, respectively.  $c_1 = (0.01L)^2$  and  $c_2 = (0.03L)^2$  where  $L = 2^8 - 1$ .

The value of SSIM is between -1 and 1 (Qasim et al., 2018), where one is for the structurally identical images. Therefore, the closer the SSIM value is to 1, the more imperceptibility is (Begum & Uddin, 2021).

Normalized Cross-Correlation (NCC) was used to evaluate the robustness of the watermarking method. First, it is computed between the extracted watermark  $W'$  and the original watermark  $W$  according to the following Equation 8 (Su et al., 2019):

$$NCC = \frac{\sum_{i=1}^M \sum_{j=1}^N W(i, j)W'(i, j)}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N W(i, j)^2} \sqrt{\sum_{i=1}^M \sum_{j=1}^N W'(i, j)^2}} \quad [8]$$

where  $M$  and  $N$  are the image dimension (Su et al., 2019). The optimal value of NCC is 1. Thus, the closer the NCC value to the optimal value, the more robust the method is (Begum & Uddin, 2021). On the other hand, Accuracy Ratio (AR) is the ratio of the correctly extracted bits  $C_b$  to the original watermark bits  $N_b$ . It can be computed as the following Equation 9 (Qasim et al., 2018):

$$AR = C_b / N_b \quad [9]$$

## RESULT AND DISCUSSION

Inclusive experiments have been performed using MATLAB to evaluate the performance of the proposed method. These experiments have been conducted on a set of both colored and grayscale standard images of size  $512 \times 512$  as host images and a binary image of size  $512/N \times 512/N$  as a watermark image where  $N = \{4, 8, 16\}$ .  $N$  also represents the dimension of the host image's squared blocks (Figure 5). Because the image's blocks are square,  $N$  will be used throughout the rest of this paper to represent block size.

The strategy of the experiments included the following sequential steps:

1. Investigate the appropriate bit planes for embedding for each block size.
2. Investigate the suitable block size among those blocks.
3. Compare the proposed method to other related works.





Figure 5. The dataset used in the experiments (a) the host images (b) the watermark image

### Experiments on Grayscale Images

The grayscale image of “Lena.bmp” was used to investigate the appropriate bit plane for embedding to fulfill the requirements of digital watermarking methods and achieve reversibility. The experiments were performed for  $N = \{4, 8, 16\}$  and  $BP = \{4, \dots, 15\}$ . PSNR and SSIM were computed between the original and watermarked images to measure the imperceptibility. Furthermore, they were computed again between the original and recovered host images to check the reversibility (Figures 6 & 7). The NCC and AR were calculated for the same  $N$  and  $BP$  values to assess the robustness (Figure 8).

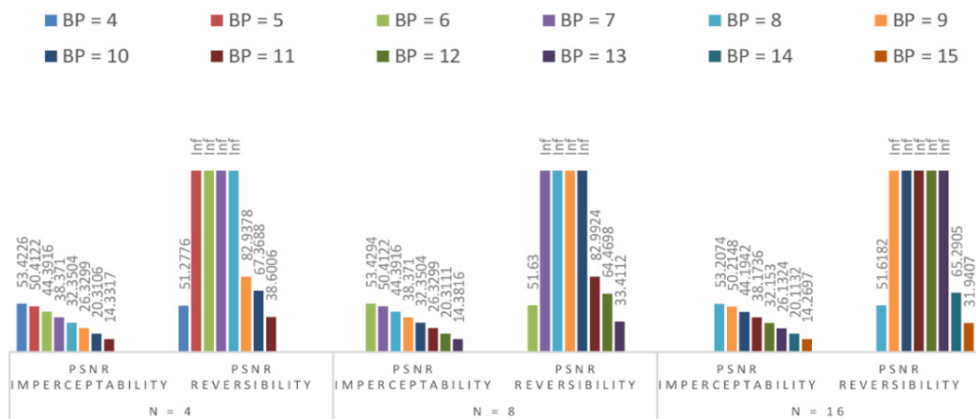


Figure 6. The PSNR of the proposed method uses different bit planes  $BP$  and block size  $N$  of the grayscale image of "Lena.bmp"

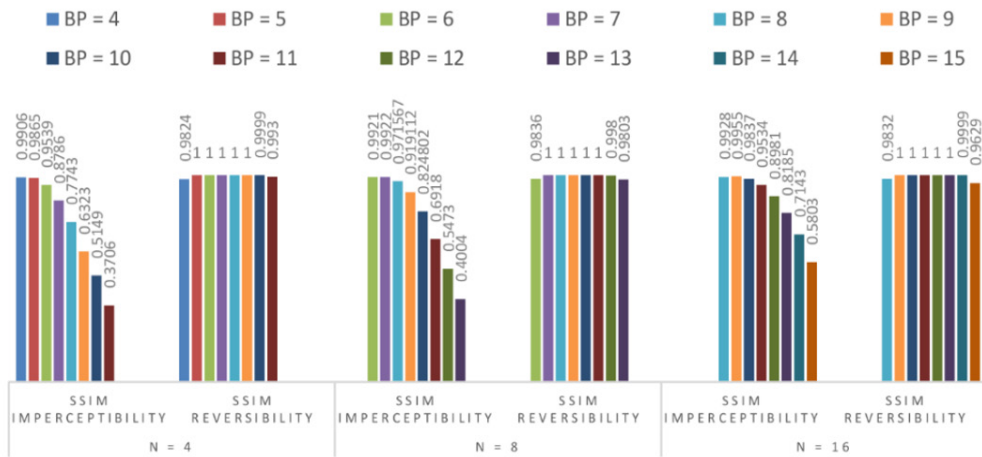


Figure 7. The SSIM of the proposed method uses different bit planes BP and block size N of the grayscale image of “Lena.bmp”

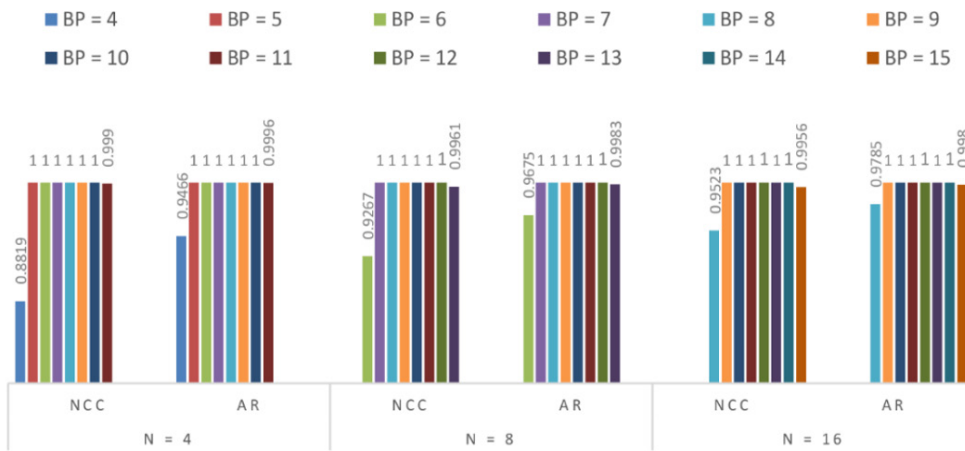


Figure 8. The robustness of the proposed method using different bit planes BP and block size N of the non-attacked grayscale image of “Lena.bmp”

Equation 10 represents the relationship between block size N and the bit plane BP.

$$BP_j = BP_i + 2 \times (\log_2(N_j) - \log_2(N_i)) \quad [10]$$

According to Equation 10, the difference between the original and the watermarked images was almost constant for different N values. The PSNR in the case of embedding in BP = 6 for N = 8 was almost the same as the PSNR of embedding in BP = 4 and BP = 8 for

$N=4$  and  $N=16$ , respectively. However, this relationship did not apply clearly to the values of SSIM. Figures 6, 7, and 8 show that, for each value of  $N$ , there is a set of bit planes  $BP$  that fulfills good imperceptibility ( $PSNR > 32$  dB) and reversibility ( $PSNR = Inf$ ) as well as full robustness when no attacks ( $NCC = 1$ ). These sets were  $BP = \{5, 6, 7, 8\}$ ,  $BP = \{7, 8, 9, 10\}$ , and  $BP = \{9, 10, 11, 12\}$  for  $N=4$ ,  $N=8$ , and  $N=16$ , respectively. Accordingly, further tests have been performed to decide the embedding bit plane among those ranges that could achieve the highest robustness against JPEG compression 50%, filtering attack (Median filter  $3 \times 3$ , and Low pass filter  $3 \times 3$ ), and noise attack (Gaussian noise, Poisson noise, Salt & pepper noise, and Speckle noise) (Figures 9, 10, & 11).

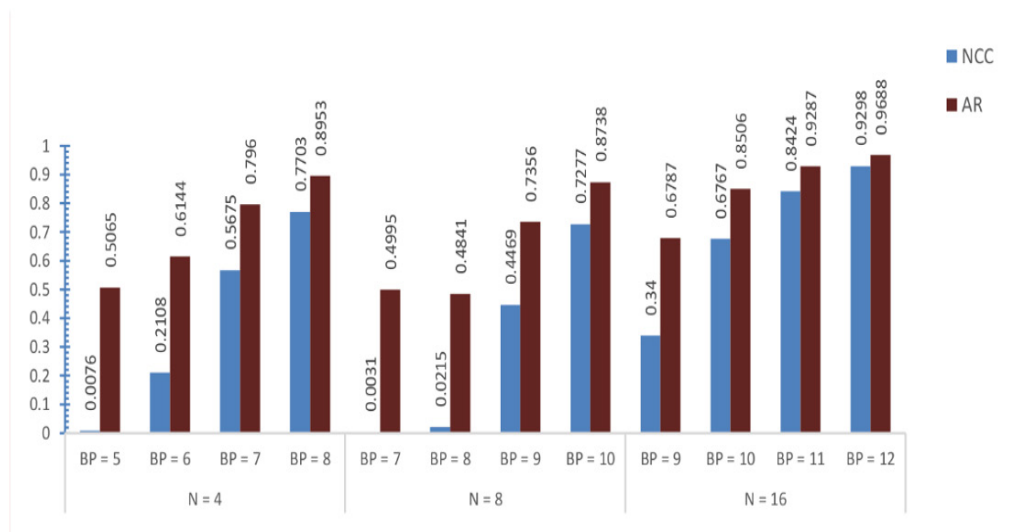


Figure 9. The robustness of the proposed method against JPEG compression 50% using the grayscale image of "Lena.bmp"

Figures 9, 10, and 11 show that the greater the value of  $BP$ , the lower the imperceptibility (Figure 12) but the higher the resistance to attacks conversely (Table 2). Therefore, the appropriate bit planes are  $BP = 8$ ,  $BP = 9$ , and  $BP = 10$  for  $N = 4$ ,  $N = 8$ , and  $N = 16$ , respectively. However, the embedding in bit plane  $BP = 9$  in the case of  $N = 8$  could be considered a suitable choice that fulfills the tradeoff between the basic digital watermarking requirements. Finally, it is worth noting that the payloads of the proposed method are 0.0625 bit/pixel, 0.0156 bit/pixel, and 0.0039 bit/pixel for  $N = 4$ ,  $N = 8$ , and  $N = 16$ , respectively.

Additional grayscale images were used to test the efficiency of the proposed method in the case of embedding in the bit plane  $BP = 9$  for block size  $N = 8$  (Table 3).

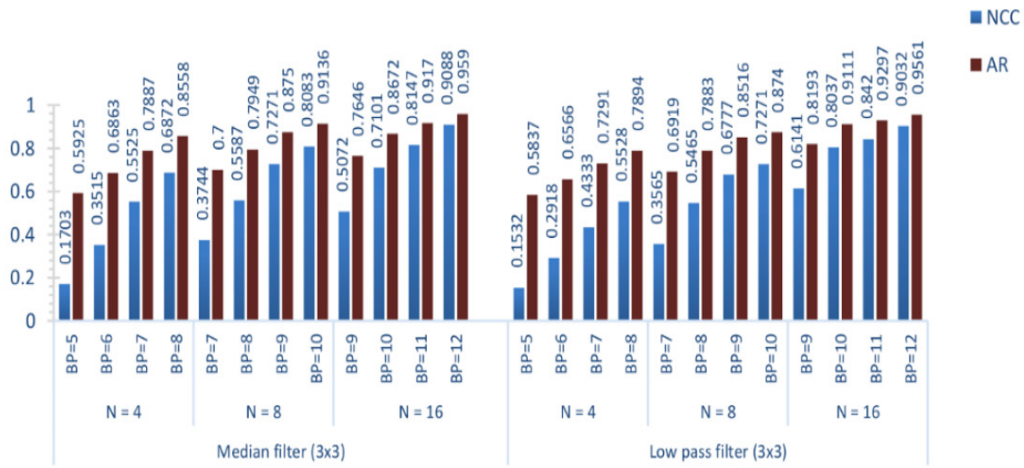
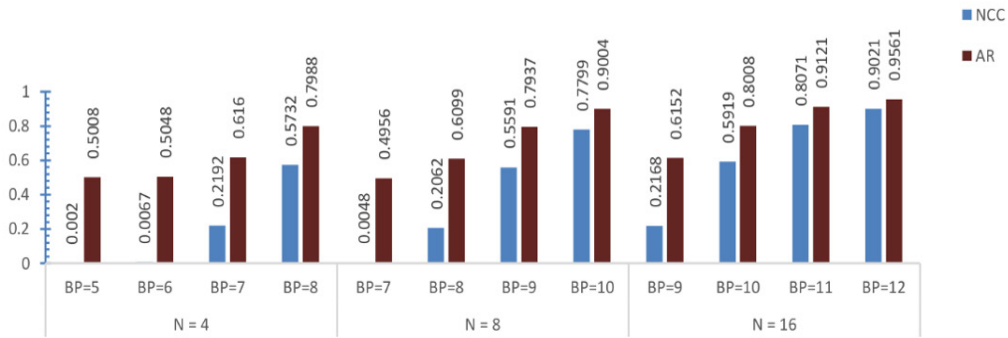
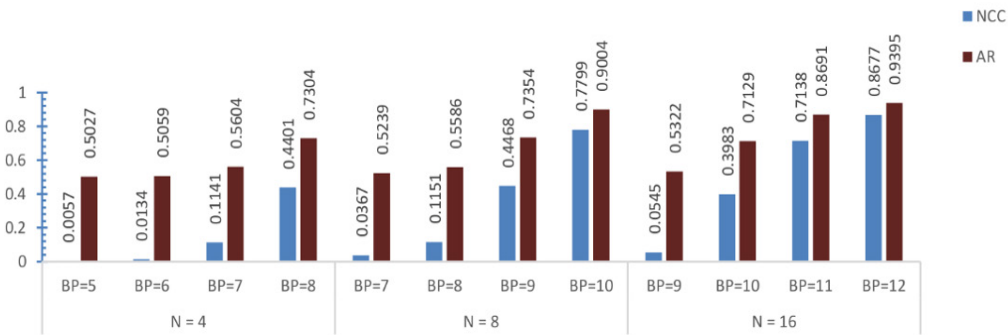


Figure 10. The robustness of the proposed method against filtering attacks using the grayscale image of “Lena.bmp”

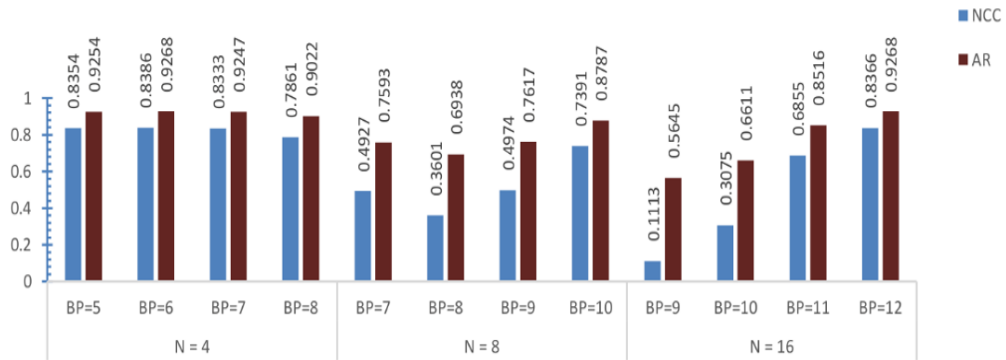


(a) Robustness against Gaussian noise 0.001

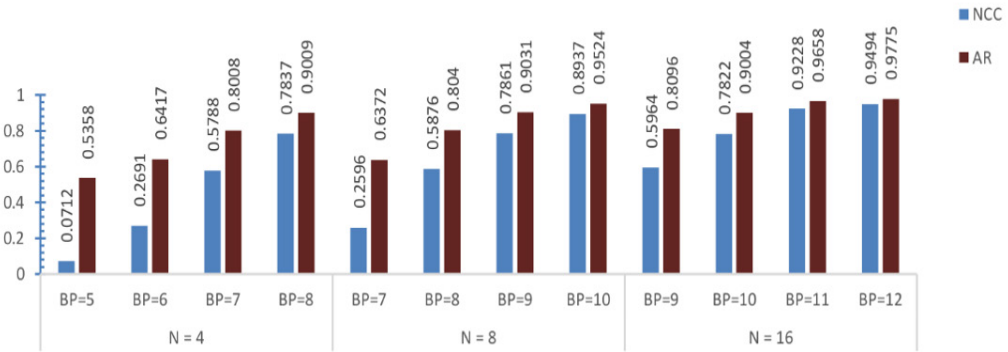


(b) Robustness against Poisson noise

Figure 11. The robustness of the proposed method against noise attacks using the grayscale image of “Lena.bmp”



(c) Robustness against salt & pepper noise 0.01



(d) Robustness against speckle noise 0.001

Figure 11. (Continue)



Figure 12. The imperceptibility of the proposed method using the grayscale image of “Lena.bmp”



Figure 12. (Continue)

Table 2

The NCC/AR of the proposed method on the attacked grayscale image of "Lena.bmp"

Attack	N = 4		N = 8		N = 16		
	BP = 7	BP = 8	BP = 9	BP = 10	BP = 10	BP = 11	BP = 12
JPEG compression 50%	 0.5675/ 0.7960	 0.7703/ 0.8953	 0.4469/ 0.7356	 0.7277/ 0.8738	 0.6767/ 0.8506	 0.8424/ 0.9287	 0.9298/ 0.9688
Gaussian noise 0.001	 0.2192/ 0.6160	 0.5732/ 0.7988	 0.5591/ 0.7937	 0.7799/ 0.9004	 0.5919/ 0.8008	 0.8071/ 0.9121	 0.9021/ 0.9561
Poisson noise	 0.1141/ 0.5604	 0.4401/ 0.7304	 0.4468/ 0.7354	 0.7799/ 0.9004	 0.3983/ 0.7129	 0.7138/ 0.8691	 0.8677/ 0.9395
Salt and Pepper noise 0.001	 0.8333/ 0.9247	 0.7861/ 0.9022	 0.4974/ 0.7617	 0.7391/ 0.8787	 0.3075/ 0.6611	 0.6855/ 0.8516	 0.8366/ 0.9268
Speckle noise 0.001	 0.5788/ 0.8008	 0.7837/ 0.9009	 0.7861/ 0.9031	 0.8937/ 0.9524	 0.7822/ 0.9004	 0.9228/ 0.9658	 0.9494/ 0.9775
Median filter	 0.5525/ 0.7887	 0.6872/ 0.8558	 0.7271/ 0.8750	 0.8083/ 0.9136	 0.7101/ 0.8672	 0.8147/ 0.9170	 0.9088/ 0.9590
Low pass filter	 0.4333/ 0.7291	 0.5528/ 0.7894	 0.6777/ 0.8516	 0.7271/ 0.8740	 0.8037/ 0.9111	 0.8420/ 0.9297	 0.9032/ 0.9561

Table 3  
The efficiency of the proposed method using grayscale images, BP =9 for N=8

Image Name	Robustness NCC / AR												
	Compression					Noise					Filtering		
	Imperceptibility	PSNR/SIM	Reversibility	No attack	JPEG 30	JPEG 60	JPEG 80	Gaussian noise 0.002	Poisson noise	Salt and Pepper noise 0.01	Speckle noise 0.001	Median filter	Low pass filter
Airplane	38.1192	Inf/1	1/1	1/1	0.5408	0.7206	0.7454	0.4031	0.3236	0.5222	0.6515/ 0.8367	0.6577	0.6032
	/	0.8419	Inf/1	1/1	0.7834	0.8716	0.8845	0.7102	0.6692	0.7695	0.8251	/	/
Barbara	38.0090	Inf/1	1/1	1/1	0.5587	0.8037	0.8914	0.4124	0.4543	0.5076	0.8251	0.6326	0.6380
	/	0.9310	Inf/1	1/1	0.7922	0.9109	0.9514	0.7156	0.7397	0.7668	/	/	/
Boat	38.1556	Inf/1	1/1	1/1	0.5074	0.7744	0.8821	0.4070	0.4188	0.4899	0.7618	0.6326	0.6371
	/	0.91101	Inf/1	1/1	0.7659	0.8967	0.9473	0.7148	0.7200	0.7568	0.8911	/	/
Elaine	38.2998	Inf/1	1/1	1/1	0.5255	0.7670/ 0.8933	0.8784	0.4012	0.4076	0.5137	0.7407	0.6961	0.6992
	/	0.9576	Inf/1	1/1	0.7749	0.9456	0.9456	0.7158	0.7148	0.7708	0.8811	0.8594	0.8604
Goldhill	38.0090	Inf/1	1/1	1/1	0.5662	0.7695	0.8872	0.4165	0.4666	0.4889	0.8082	0.6255	0.6586
	/	0.9585	Inf/1	1/1	0.7944	0.8938	0.9492	0.7202	0.7446	0.7544	0.9124	0.8245	0.8398
House	38.0459	Inf/1	1/1	1/1	0.5432	0.7380	0.8607	0.4226	0.3479	0.4945	0.7069	0.6436	0.6156
	/	0.9020	Inf/1	1/1	0.7820	0.8804	0.9370	0.7224	0.6858	0.7590	0.8638	0.8374	0.8220
Sailboat	38.4416	Inf/1	1/1	1/1	0.5766	0.7440	0.7369	0.3793	0.4229	0.4988	0.6927	0.5363	0.5378
	/	0.9240	Inf/1	1/1	0.7986	0.8833	0.8787	0.7021	0.7253	0.7639	0.8596	0.7795	0.7798

The efficiency of the proposed method in the case of embedding in bit planes BP = 9 and BP =13 for N = 8 was compared with the bit plane embedding method proposed by Ahmed & Moskowitz (2004) (Figure 13). Despite the high capacity and imperceptibility of the method proposed by Ahmed and Moskowitz (2004), which involved all the magnitudes in the embedding process, it showed low robustness compared to the current method. The proposed method utilized only the DC coefficients in the embedding. Furthermore, according to the proposed method, embedding in the 9<sup>th</sup> bit plane showed higher imperceptibility and robustness than the embedding in the 13<sup>th</sup> bit plane, according to Ahmed and Moskowitz (2004).

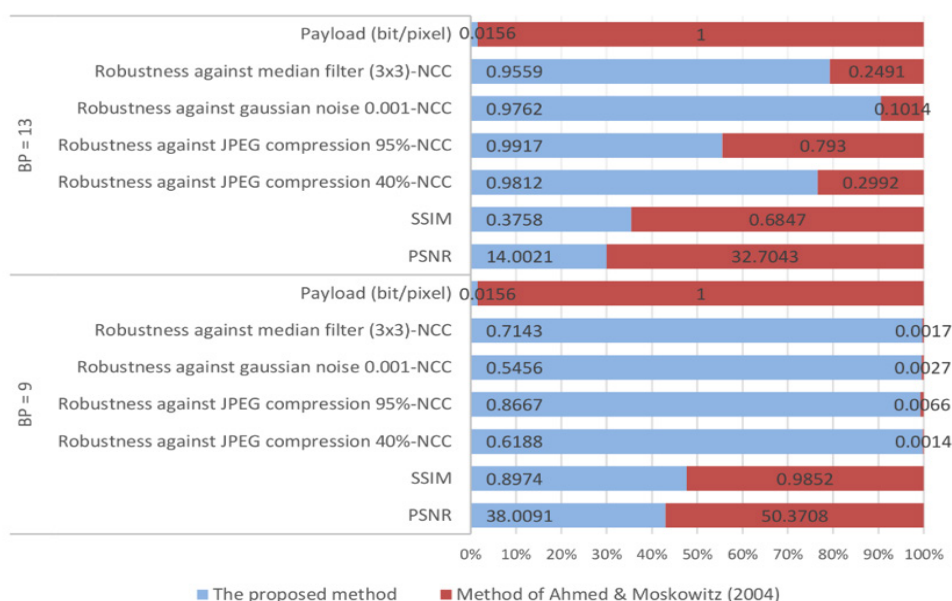


Figure 13. The comparison between the proposed method and the method of Ahmed and Moskowitz (2004)

### Experiments on Colored Images

For the colored images, the same steps of the proposed method were carried out on the luminance components (Y) after converting the host image into YUV color space. The previous experiments were performed again on the colored image of “Lena.bmp” to investigate the appropriate bit planes for embedding (Figures 14, 15, & 16). It should be pointed out that the PSNR of the colored images was computed for only the luminance channel.

Figures 14 and 15 show that reversibility was not achieved for the colored images using the embedding into any bit plane (PSNR ≠ Inf) regardless of the block size despite the high



corresponding SSIM values. However, the set of bit planes that fulfill good imperceptibility and some levels of robustness (Figure 16) were  $BP = \{5, 6, 7, 8\}$ ,  $BP = \{7, 8, 9, 10\}$ ,  $BP = \{9, 10, 11, 12\}$  for  $N = 4$ ,  $N = 8$ , and  $N = 16$ , respectively.

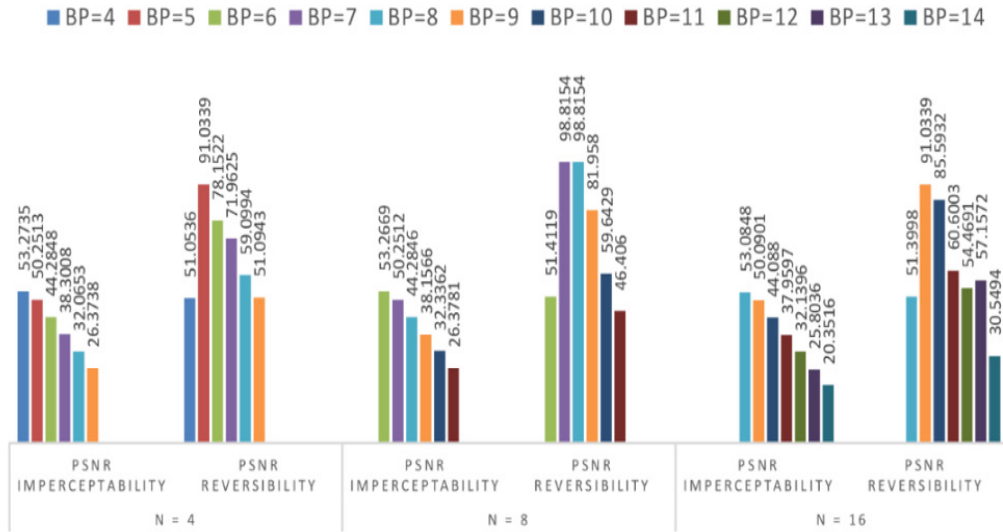


Figure 14. The PSNR of the proposed method using different bit planes BP and block size N of the colored image of “Lena.bmp”

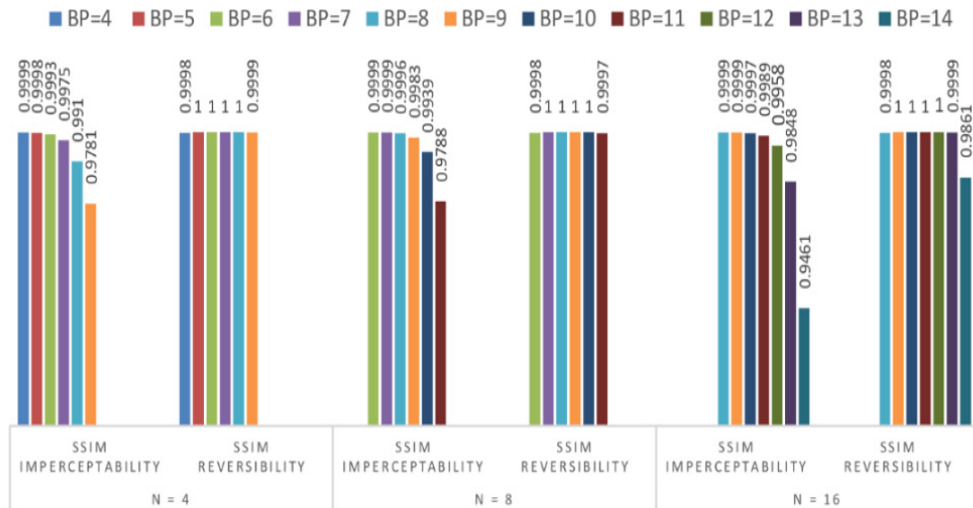


Figure 15. The SSIM of the proposed method using different bit planes BP and block size N of the colored image of “Lena.bmp”

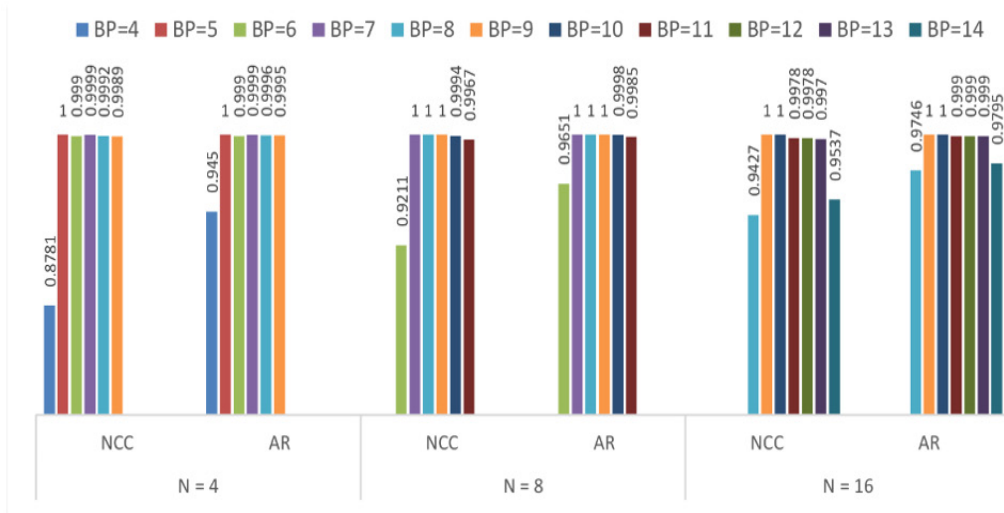


Figure 16. The robustness of the proposed method using different bit planes BP and block size N of the non-attacked colored image of “lena.bmp”

Figures 17, 18, and 19 show the robustness of the proposed method against attacks. In addition, Figures 17, 18, and 19 reveal that the bit planes BP = 8, PB = 9, BP = 10 for N = 4, N = 8, and N = 16, respectively, could be used for semi-fragile watermarking depending on the system requirement (Table 4) (Figure 20).

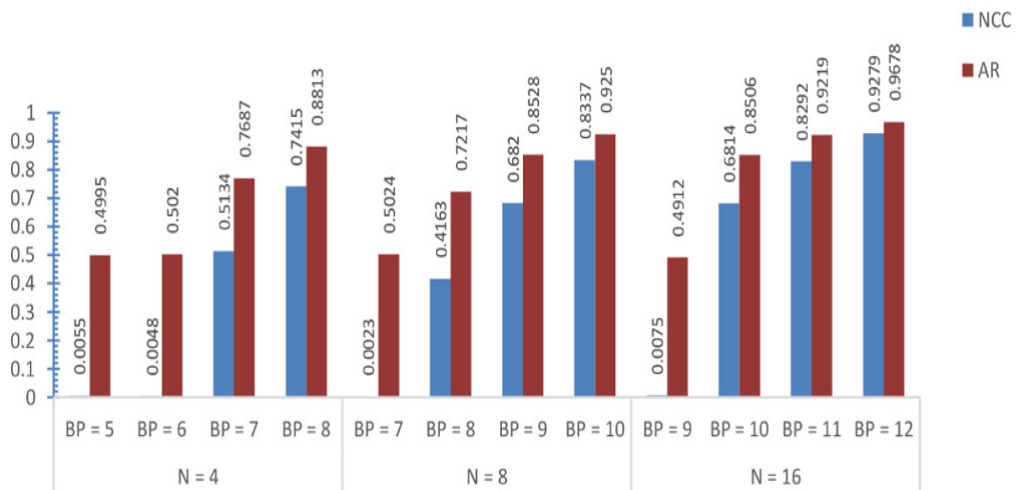


Figure 17. The robustness of the proposed method against JPEG compression 50% using the colored image of “Lena.bmp”

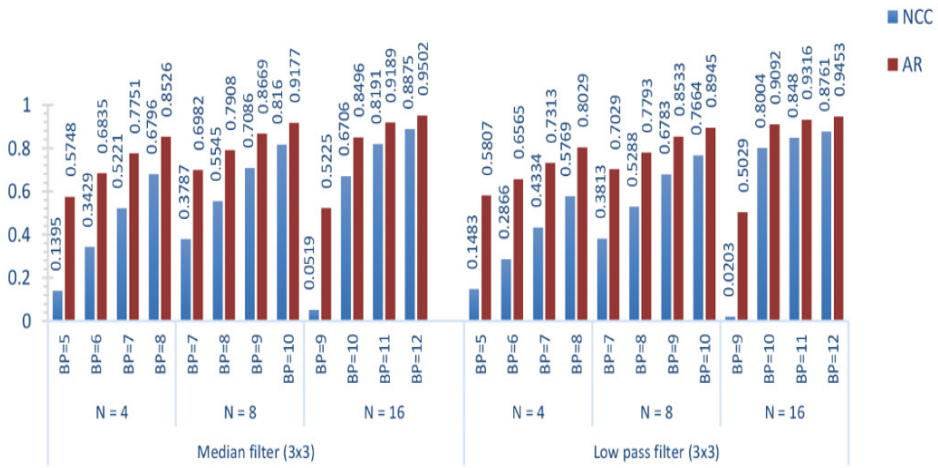
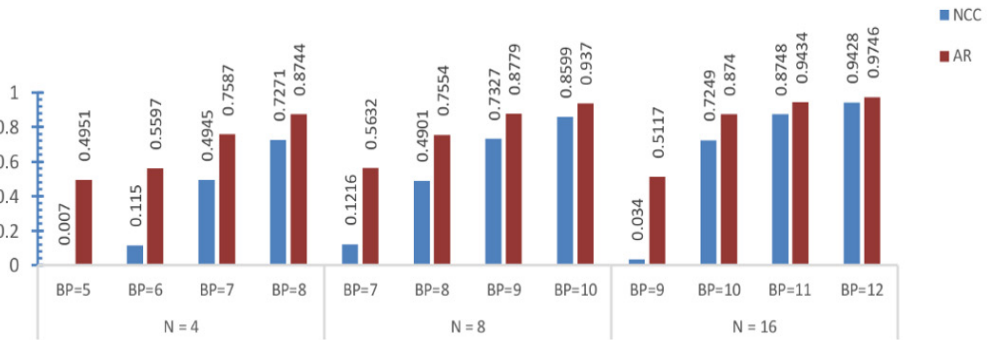
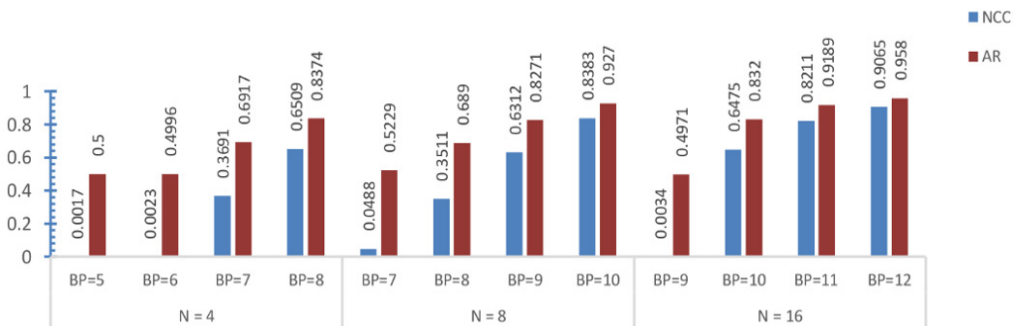


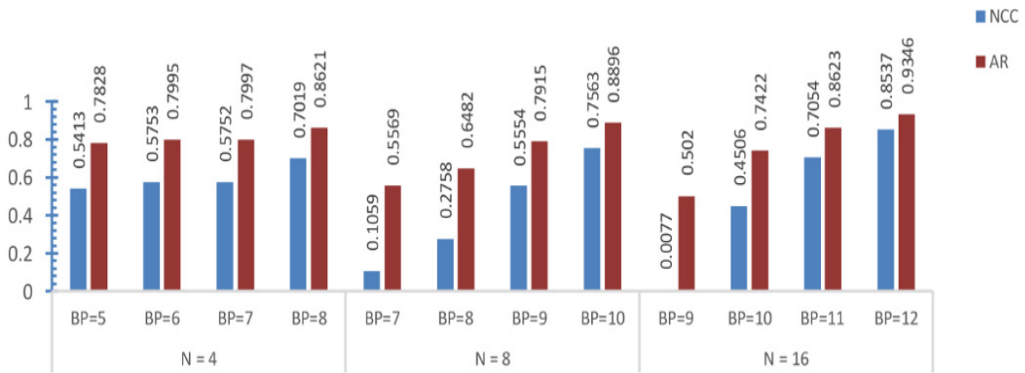
Figure 18. The robustness of the proposed method against filtering attacks using the colored image of “Lena. bmp”



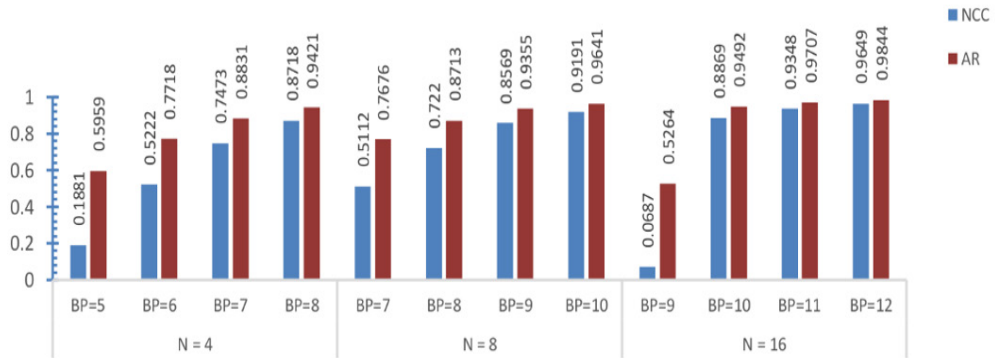
(a) Robustness against Gaussian noise 0.001



(b) Robustness against Poisson noise



(c) Robustness against salt & pepper noise 0.01



(d) Robustness against speckle noise 0.001

Figure 19. The robustness of the proposed method against noise attacks using the colored image of “Lena.bmp”

Table 4

The NCC/AR of the proposed method on the attacked colored image of "Lena.bmp"

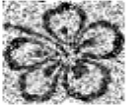
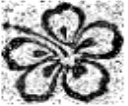















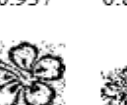
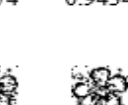




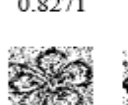
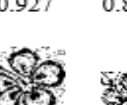
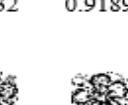




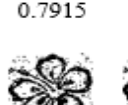
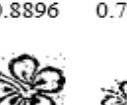
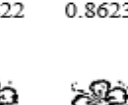




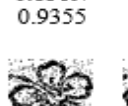
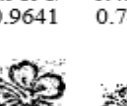
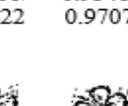





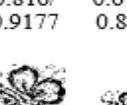
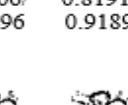


Attack	N = 4		N = 8		N = 16		
	BP = 7	BP = 8	BP = 9	BP = 10	BP = 10	BP = 11	BP = 12
JPEG compression 50%	 0.5134/ 0.7687	 0.7415/ 0.8813	 0.682/ 0.8528	 0.8337/ 0.925	 0.6814/ 0.8506	 0.8292/ 0.9219	 0.9279/ 0.9678
Gaussian noise 0.001	 0.4945/ 0.7587	 0.7271/ 0.8744	 0.7327/ 0.8779	 0.8599/ 0.937	 0.7249/ 0.874	 0.8748/ 0.9434	 0.9428/ 0.9746
Poisson noise	 0.3691/ 0.6917	 0.6509/ 0.8374	 0.6312/ 0.8271	 0.8383/ 0.927	 0.6475/ 0.832	 0.8211/ 0.9189	 0.9065/ 0.958
Salt and Pepper noise 0.001	 0.5752/ 0.7997	 0.7019/ 0.8621	 0.5554/ 0.7915	 0.7563/ 0.8896	 0.4506/ 0.7422	 0.7054/ 0.8623	 0.8537/ 0.9346
Speckle noise 0.001	 0.7473/ 0.8831	 0.8718/ 0.9421	 0.8569/ 0.9355	 0.9191/ 0.9641	 0.4506/ 0.7422	 0.9348/ 0.9707	 0.9649/ 0.9844
Median filter	 0.5221/ 0.7751	 0.6796/ 0.8526	 0.7086/ 0.8669	 0.816/ 0.9177	 0.6706/ 0.8496	 0.8191/ 0.9189	 0.8875/ 0.9502
Low pass filter	 0.4334/ 0.7313	 0.5769/ 0.8029	 0.6783/ 0.8533	 0.7664/ 0.8945	 0.8004/ 0.9092	 0.848/ 0.9316	 0.8761/ 0.9453



Figure 20. The imperceptibility of the proposed method using the colored image of “Lena.bmp”

Additional colored images were used to test embedding efficiency in the bit plane BP = 9 for block size N = 8 (Table 5).

The efficiency of the proposed method was compared with both methods of Zhang et al. (2020) and Su et al. (2019) (Figure 21). In this comparison, the embedding was performed on the colored image of “Lena.bmp” in RGB color space. The watermark was embedded in the DC coefficients of red, green, and blue components in the bit plane BP = 8, and BP=10 for block sizes N = 4, and N = 8, respectively. Despite the high PSNR of Zhang et al. (2020) and Su et al. (2019), the proposed method outperformed these two methods in terms of SSIM values regardless of the payload. It means that the distortion in the structural contents of the image due to embedding was less in the currently proposed method. Figure 21 shows that both Zhang et al. (2020) and Su et al. (2019) methods were more robust against JPEG compression. In contrast, the level of robustness against salt & pepper noise was very close in all the methods, despite the high payload of the proposed method in the case of N = 4, which reached double the payload of both Zhang et al. (2020) and Su et al. (2019) methods. On the other hand, the robustness of the median filter was less robust in the proposed method when N = 4.

Table 5  
The efficiency of the proposed method using colored images, BP = 9 for N=8

Image Name	Robustness NCC / AR											
	Compression					Noise					Filtering	
	Imperceptibility PSNR/SSIM	Reversibility PSNR/SSIM	No attac	JPEG 30	JPEG 60	JPEG 80	Gaussian noise 0.002	Poisson noise	Salt and Pepper noise 0.01	Speckle noise 0.001	Median filter	Low pass filter
Airplane	37.8621 /	65.2868 /	0.9994 /	0.4852 /	0.7562 /	0.8327 /	0.6433 /	0.5698 /	0.5171 /	0.8160 /	0.6969 /	0.6533 /
	0.9388	0.9999	0.9998	0.7571	0.8867	0.9243	0.8335	0.7961	0.7676	0.9153	0.8606	0.8408
Barbara	37.8222 /	98.6221 /	1/1	0.5058 /	0.7662 /	0.8665 /	0.6369 /	0.6723 /	0.5542 /	0.8964 /	0.6423 /	0.6318 /
	0.9935	1		0.7676	0.8933	0.9399	0.8306	0.8464	0.7876	0.9539	0.8333	0.8279
House	37.9797 /	59.3828 /	0.9978 /	0.4985 /	0.7322 /	0.8447 /	0.5566 /	0.5212 /	0.5214 /	0.7690 /	0.6916 /	0.6774 /
	0.9922	1	0.9990	0.7617	0.8774	0.9299	0.7900	0.7722	0.7727	0.8945	0.8562	0.8486
Mandrill	37.7094 /	75.4084 /		0.4956 /	0.7550 /	0.8587 /	0.6341 /	0.6302 /	0.5771 /	0.8629 /	0.3306 /	0.5105 /
	0.9975	1	1/1	0.7644	0.8889	0.9365	0.8308	0.8289	0.8025	0.9287	0.6665	0.7703
Sailboat	38.0943 /	60.4602 /		0.5052 /	0.7530 /	0.8667 /	0.6117 /	0.6455 /	0.5347 /	0.8604 /	0.5850 /	0.6122 /
	0.9948	1	1/1	0.7646	0.8875	0.9402	0.8186	0.8342	0.7834	0.9373	0.7998	0.8186
Koala	38.7242 /	71.6941 /		0.5058 /	0.7350 /	0.8708 /	0.6199 /	0.6804 /	0.5605 /	0.8887 /	0.6547 /	0.6582 /
	0.9911	0.9999	1/1	0.7642	0.8779	0.9424	0.8193	0.8528	0.7917	0.9502	0.8401	0.8420
Goldhill	38.7394 /	60.7647 /	0.9967 /	0.5493 /	0.6083 /	0.8596 /	0.5098 /	0.5218 /	0.5526 /	0.7607 /	0.5727 /	0.6341 /
	0.9088	0.9972	0.9985	0.7915	0.8188	0.9370	0.7603	0.7627	0.7886	0.8877	0.8020	0.8313

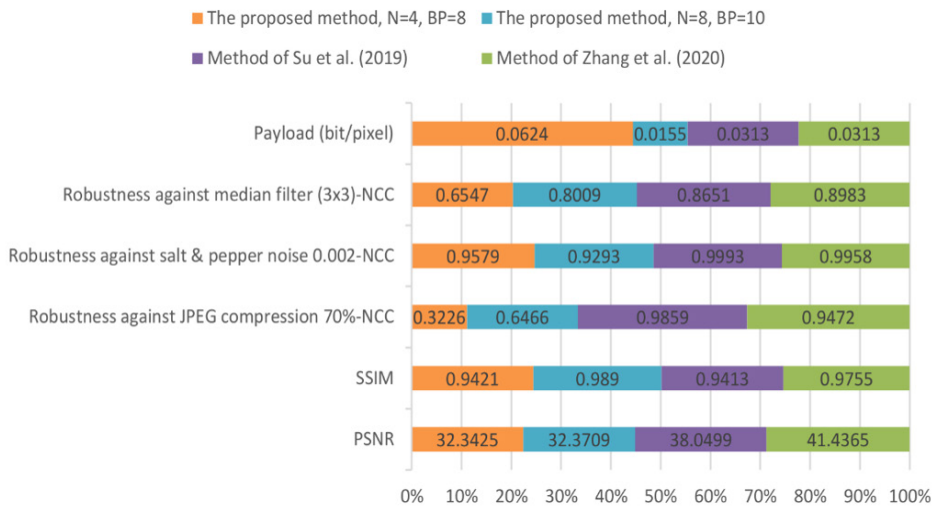


Figure 21. The comparison between the proposed method and the methods of Zhang et al. (2020) and Su et al. (2019)

## CONCLUSION

This paper proposed a semi-fragile, semi-blind watermarking method for image ownership protection. The watermark was embedded in a middle bit plane of the DC coefficients of the DFT magnitudes. The proposed method used a combination of double Feynman and XOR reversible gates to shuffle the watermark with the secret key and original bit plane of the DC coefficients to improve the security level and ensure reversibility. The experimental results on grayscale and colored images show that the greater the bit plane of DC coefficients, the more robust but less imperceptible the watermarking method is. However, the best bit planes used for semi-fragile embedding were BP = 8, PB = 9, BP = 10 for block sizes N = 4, N = 8, and N = 16, respectively. Compared to related works, it was discovered that utilizing only the DC coefficients rather than all the DFT magnitudes has a significant impact on increasing robustness. In addition, embedding a watermark in the DC coefficients in the frequency domain rather than the spatial domain reduces the structural content’s distortion of the image. Furthermore, the proposed method using grayscale images is particularly effective in applications where reversibility is desired. However, further studies are recommended to find a reversible embedding method for colored images.

## ACKNOWLEDGEMENT

The authors acknowledge the support of the University of Mosul for this manuscript completion.



## REFERENCES

- Ahmed, F., & Moskowitz, I. S. (2004). Correlation-based watermarking method for image authentication applications. *Optical Engineering*, 43(8), 1833-1838. <https://doi.org/10.1117/1.1763589>
- Ahmed, F., & Moskowitz, I. S. (2006). A semi-reversible watermark for medical image authentication. In *1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare* (pp. 59-62). IEEE Publishing. <https://doi.org/10.1109/DDHH.2006.1624797>
- Ansari, A., Hong, S., Saavedra, G., Javidi, B., & Martinez-Corral, M. (2018). Ownership protection of plenoptic images by robust and reversible watermarking. *Optics and Lasers in Engineering*, 107, 325-334. <https://doi.org/10.1016/j.optlaseng.2018.03.028>
- Begum, M., & Uddin, M. S. (2021). Implementation of secured and robust DFT-based image watermark through hybridization with decomposition algorithm. *SN Computer Science*, 2, Article 221. <https://doi.org/10.1007/s42979-021-00608-6>
- Feng, B., Li, X., Jie, Y., Guo, C., & Fu, H. (2019). A novel semi-fragile digital watermarking scheme for scrambled image authentication and restoration. *Mobile Networks and Applications*, 25, 82-94. <https://doi.org/10.1007/s11036-018-1186-9>
- Jimson, N., & Hemachandran, K. (2018). DFT based digital image watermarking: A survey. *International Journal of Advanced Research in Computer Science*, 9(2), 540-544. <https://doi.org/10.26483/ijarcs.v9i2.5747>
- Khalilidan, S., Mahdavi, M., Balouchestani, A., Moti, Z., & Hallaj, Y. (2020) A semi-blind watermarking method for authentication of face images using autoencoders. In *2020 6th International Conference on Web Research (ICWR)* (pp. 229-233). IEEE Publishing. <https://doi.org/10.1109/ICWR49608.2020.9122276>
- Krishna, K. B., & Ramesh, A. P. (2019). Implementation of sequential circuit using feynman and fredkin reversible logic gates. In *Journal of Physics: Conference Series (Vol. 1228, No. 1, p. 012047)*. IOP Publishing. <https://doi.org/10.1088/1742-6596/1228/1/012047>
- Luo, Y., Li, L., Liu, J., Tang, S., Cao, L., Zhang, S., Qiu, S., & Cao, Y. (2021). A multi-scale image watermarking based on integer wavelet transform and singular value decomposition. *Expert Systems with Applications*, 168, Article 114272. <https://doi.org/10.1016/j.eswa.2020.114272>
- Menendez-Ortiz, A., Feregrino-Uribe, C., Hasimoto-Beltran R., & Garcia-Hernandez, J. J. (2019). A survey on reversible watermarking for multimedia content: A robustness overview. *IEEE Access*, 7, 132662-132681. <https://doi.org/10.1109/ACCESS.2019.2940972>
- Qasim, A. F, Meziane, F., & Aspin, R. (2018). Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review. *Computer Science Review*, 27, 45-60. <https://doi.org/10.1016/j.cosrev.2017.11.003>
- Setiadi, D. R. I. (2021). PSNR vs SSIM: Imperceptibility quality assessment for image steganography. *Multimedia Tools and Applications*, 80, 8423-8444. <https://doi.org/10.1007/s11042-020-10035-z>
- Su, Q., Liu D., Yuan, Z., Wang, G., Zhang, X., Chen, B., & Yao, T. (2019). New rapid and robust color image watermarking technique in spatial domain. *IEEE Access*, 7, 30398-30409. <https://doi.org/10.1109/ACCESS.2019.2895062>

- Zhang, X., Su, Q., Yuan Z., & Liu, D. (2020). An efficient blind color image watermarking algorithm in spatial domain combining discrete Fourier transform. *Optik*, 219, Article 165272. <https://doi.org/10.1016/j.ijleo.2020.165272>
- Zhu, X., Wu, J., & Sang, J. (2007). On the fragility of the binary phase-only filter based digital image watermarking. In *MIPPR 2007: Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications* (Vol. 6790, pp. 1367-1372). SPIE Publishing. <https://doi.org/10.1117/12.751141>.