

Contents lists available online at [TALENTA Publisher](https://talenta.publisher.talenta.com)

## DATA SCIENCE: JOURNAL OF COMPUTING AND APPLIED INFORMATICS (JoCAI)

Journal homepage: <https://jocai.usu.ac.id>

# Analysis of Embedding Locations in the Subband Frequency DCT on Scanned Images

Indri Hidayati<sup>1</sup>, Mohammad Andri Budiman<sup>2,\*</sup>, Muhammad Zarlis<sup>3</sup><sup>1,2</sup>*Master of Informatics Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia*<sup>3</sup>*Department of Information Systems, Bina Nusantara University, Jakarta, Indonesia*

### ARTICLE INFO

#### Article history:

Received 30 November 2022

Revised 26 January 2023

Accepted 27 January 2023

Published online 30 January 2023

#### Keywords:

Discrete Cosine Transform Invisible

Watermark

PSNR

SSIM

Subband DCT

Watermarking Technique

#### Email:

<sup>1</sup>[indrihidayati8@gmail.com](mailto:indrihidayati8@gmail.com)<sup>2</sup>[mandrib@usu.ac.id](mailto:mandrib@usu.ac.id)<sup>3</sup>[muhammad.zarlis@binus.edu](mailto:muhammad.zarlis@binus.edu)

#### Corresponding Author:

Mohammad Andri Budiman

### ABSTRACT

Uploading an identity card as an image for the account verification process or transactions online can be a threat to application users. Identity card theft can be carried out by irresponsible persons if the application can be hacked. Therefore, protection of the image is required for authentication. In this study, the proposed technique is watermarking. A watermark in the form of a binary image will be embedded into the image as ownership using a Discrete Cosine Transform. The Discrete Cosine Transform works in the frequency domain. The location of the embedding of different watermarks was analysed in each 8×8 DCT block. The results of the analysis to assess the imperceptibility of original images and watermarked images using PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity Index Measure), while assessing the watermark robustness embedded using NCC (Normalized Cross Correlation). The results show PSNR (Peak Signal to Noise Ratio)  $\geq 54$  dB with a watermark strength of 0,1 and an average SSIM (Structural Similarity Index Measure)  $\geq 0,9$  on 4 scanned images in BMP format with a resolution of 100 DPI. A good watermark embedding is done on the green component at middle frequencies to maintain a balance between imperceptibility and robustness. In contrast, the red component at low frequency is vulnerable to attacks in the form of brightness +20 and contrast +50 with an average NCC (Normalized Cross Correlation)  $\leq 0,85$ .

#### IEEE style in citing this article:

I. Hidayati, M.A. Budiman and M. Zarlis, " Analysis of Embedding Locations in the Subband Frequency DCT on Scanned Images." *Data Science: Journal of Computing and Applied Informatics (JoCAI)*, Vol. 7, No. 1, pp. 1-14, 2023.

## 1. Introduction

Currently, the dissemination of identity card information has been widespread in line with the development of information technology for account verification or online transactions. Weaknesses in the system can threaten users who are not guaranteed the security and protection of user data. Identity card theft can occur if an irresponsible person can hack a user's account.

Many techniques have been proposed to protect images, such as cryptography, watermarking, digital signatures, and so on [1]. In this study, the proposed technique is watermarking. Digital watermarking embeds information into digital images to verify ownership and authenticity [2]. Image watermarking techniques can be divided based on spatial and frequency domains. The spatial domain works directly on the pixels, while the frequency domain works on the image transformation coefficient [3].

In this study, the proposed watermark embedding works in the frequency domain using Discrete Cosine Transform (DCT). The Discrete Cosine Transform (DCT) converts the signal into fundamental frequency components, representing the image as a sum of sinusoids of various magnitudes and frequencies. Discrete Cosine Transform (DCT) segmented non-overlapping image blocks and applied them to each block resulting in three frequency subbands, namely low, middle, and high frequencies [3].

Several watermark embedding methods have been proposed in the last five years. In 2017, embedding a watermark using the DCT method in the Y-component at high frequency by setting the coordinate (7,7) as the embedding location resulted in a PSNR of 40,82 dB [4]. The combination of watermarking techniques on medical images uses Fast Discrete Curvelet Transform (FDCuT) and Discrete Cosine Transform (DCT) with different embedding factor scales. The results showed the highest PSNR with a scale factor of  $\alpha=2$  and the highest NCC with a scale factor of  $\alpha=8$ , indicating that the higher the scale factor value will increase robustness and, vice versa, will reduce imperceptibility performance [5]. The analysis of copyright concealment techniques using DCT transformation was carried out on RGB images in JPG and BMP formats. The results show the highest average PSNR value with the BMP format [6].

An essential parameter of watermarking is determining the location of the watermark embedding on the original image [7]. Few previous studies have implemented different embedding locations in each  $8 \times 8$  DCT block. Therefore, this research was conducted to select other embedding locations in the low, middle, and high frequencies based on the cosine graph with the functional domain  $-1 \leq x \leq 1$ , set of principle values ( $0 \leq y \leq \pi$ ) where representing the frequency wave. Sorting on the AC DCT coefficient in each frequency subband is applied to obtain the coordinates of the selected pixels. This study aimed to analyze the location of the watermark embedding in the Discrete Cosine Transform frequency subband on the scanned image. The embedding technique and the selection of the embedding coefficient greatly determine the imperceptibility and robustness qualities [8].

## 2. Research Methods

In this study, there are two processes: embedding and watermark extraction. The proposed method is a watermarking technique in the frequency domain using a Discrete Cosine Transform (DCT) with a block size of  $8 \times 8$ . Watermark embedding and extraction are performed on each colour component (RGB) and frequency subband. AC DCT coefficients are sorted for different embedding locations in each  $8 \times 8$  block based on the frequency subband. The type of image proposed is a colour image (RGB) for the original image and a binary image for the watermark image.

### 2.1. Watermark Embedding

The proposed watermark embedding is shown in Figure 1.

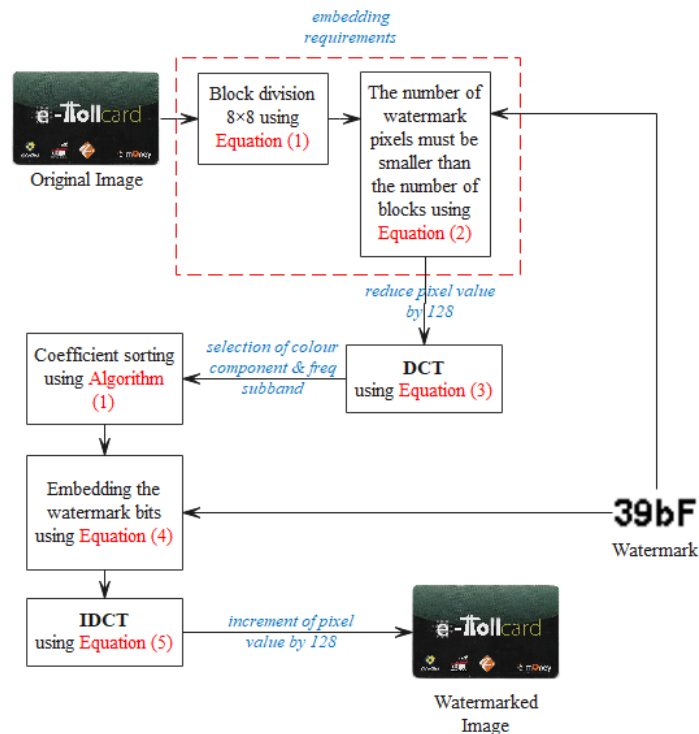


Figure 1. The proposed watermark embedding

The conditions that must be met by an original image in the watermark embedding process using Discrete Cosine Transform (DCT) are stated in Equation (1).

$$W_{max} = \left\lfloor \frac{M}{8} \right\rfloor \times \left\lfloor \frac{N}{8} \right\rfloor \quad (1)$$

Where  $W_{max}$  is the maximum number of  $8 \times 8$  blocks that will be embedded with watermark bits,  $M$  is the side length of the original image, and  $N$  is the side width of the original image. From Equation (1), it can be concluded that the requirements of a watermark can be embedded into the original image if it meets Equation (2).

$$M_w \times N_w \leq W_{max} \quad (2)$$

$M_w$  is the length of the side of the watermark image and  $N_w$  is the width of the side of the watermark image. Once the embedding conditions are met, the original image is divided into  $8 \times 8$  sized blocks across all colour component (RGB). The original image colour intensity value is obtained at this stage, and 128 is reduced [9]. Then, every  $8 \times 8$  block in all colour components (RGB) subtracted by 128 is transformed by Equation (3) to obtain the DCT coefficient.

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cdot \cos \left[ \frac{(2x+1)u\pi}{16} \right] \cdot \cos \left[ \frac{(2y+1)v\pi}{16} \right] \quad (3)$$

$u = 0, 1, 2, \dots, 7$  and  $v = 0, 1, 2, \dots, 7$ , where the magnitude of  $C(k)$  is  $\frac{1}{\sqrt{2}}$  for  $k = 0$  and 1 for the others.  $f(x, y)$  is the colour intensity value in the original image, which has been reduced by 128 and  $\pi = 180^\circ$ . The DCT coefficient is divided into three frequency subbands: low, middle, and high frequencies [10]. The distribution of the proposed DCT coefficient is similar to that of previous studies [11]. Declaration of the frequency subband variable and initialized in the form of pixel coordinates as shown in Table 1, except that the DC coefficient (0,0) is not used to maintain good visual quality [2].

Table 1. Pixel coordinates in each proposed frequency subband

Frequency Subband	Coordinates ( $u, v$ )
Low	((0,1),(1,0),(2,0),(1,1),(0,2))
Middle	((0,3),(1,2),(2,1),(3,0),(4,0),(3,1),(2,2),(1,3),(0,4),(0,5),(1,4),(2,3), (3,2),(4,1),(5,0),(6,0),(5,1),(4,2),(3,3),(2,4),(1,5),(0,6))
High	((0,7),(1,6),(2,5),(3,4),(4,3),(5,2),(6,1),(7,0),(7,1),(6,2),(5,3),(4,4), (3,5),(2,6),(1,7),(2,7),(3,6),(4,5),(5,4),(6,3),(7,2),(7,3),(6,4),(5,5), (4,6),(3,7),(4,7),(5,6),(6,5),(7,4),(7,5),(6,6),(5,7),(6,7),(7,6),(7,7))

In the next stage, the AC DCT coefficient sorting process is carried out in three steps: taking the AC DCT coefficient on the colour component (RGB) and frequency subband, sorting the AC DCT coefficient in ascending order (from smallest to largest), and fetching the selected index as the embedding location shown in Algorithm 1.

In Algorithm 1, the declaration of the index variable with the initial value = 0 because first element in an array using 0 index, where index < the total number of elements in the array is the AC DCT coefficient at the pixel coordinates that have been declared previously based on the frequency subband. The index is the sequential number of the AC DCT coefficients that are not sequential so that the AC DCT coefficients are sorted from the smallest to the largest. Taking the AC DCT coefficient index sorted by the rules of the AC DCT coefficient for low frequencies, is -1 (lowest in the subband). The AC DCT coefficient for middle frequencies is 0 or close to 0, and the AC DCT coefficient for high frequencies is 1 (highest in the subband). Therefore, the method of taking AC DCT coefficients for low frequencies at the first index. Middle frequencies with the condition that all negative AC DCT coefficients are changed to positive and sorted from smallest to largest so that the first index is taken because it is close to the value 0. High frequency on the last index with the condition that the number of indexes or the AC DCT coefficient is reduced by 1, because if there are  $n$  elements in a list, the last element is  $n - 1$ .

**Algorithm 1.** Sorting the AC DCT coefficient values in each subband to embed the watermark bits

---

```

1   INPUT      : AC DCT coefficient values in each subband
2   OUTPUT     : Selected pixel coordinates to embed watermark bits
3   SET position to zero
4   SET an empty dictionary (int(KEY), double(VALUE)) variable sort1, sort2
5   FOR i ← zero to (length of coordinate)-1 DO
6       IF layer ← red THEN
7           Add element (i(KEY), transform_coordinate(VALUE), 0) in sort1
8       ELSE IF layer ← green
9           Add element (i(KEY), transform_coordinate(VALUE), 1) in sort1
10      ELSE Add element (i(KEY), transform_coordinate(VALUE), 2) in sort1
11      END IF
12      Increment i by 1
13  END FOR
14  FOREACH result in sort1 DO
15      Sorts the elements of a sequence in ascending order according to a key ←
16      key.Value[]
17      Add element (result.Key[], result.Value[]) in sort2
18  END FOREACH
19  SET index to zero
20  SET a string variable frequency
21  FOREACH result in sort2 DO
22      IF frequency ← “low”
23          IF index ← zero THEN
24              position ← get result.Key[]
25              break when index reaches zero
26          END IF
27      END IF
28      IF frequency ← “middle”
29          FOREACH result1 in sort1 DO
30              Sorts the elements of a sequence in ascending order according to a key
31              ← key.Value[]
32              Returns the absolute value of a double precision floating-point number
33              position ← get result1.Key[]
34              break when result1.Key[] to equal zero
35          END FOREACH
36          break foreach loop
37      END IF
38      IF frequency ← “high”
39          IF index ← return total number of elements in sort2 -1 THEN
40              position ← get result.Key[]
41          END IF
42      END IF
43  END FOREACH

```

---

Then, the value of the colour intensity of the watermark image is extracted. In this study, when a binary image is displayed in the form of an image, it will be worth 255 for white and 0 for black. The value 255 is changed to value one, and the value 0 remains. The next step is to modify the AC DCT coefficient selected from the sorting results with Equation (4).

$$F'(u, v) = F(u, v) + \alpha(w) \quad (4)$$

Where  $F(u, v)$  is the DCT coefficient of the sorted result,  $\alpha$  is the strength of the watermark that adjusts the tradeoff between imperceptibility and robustness [12],  $w$  is one-bit binary data (0 or 1), where the bit value 0 becomes -1, and the bit is one fixed. The decrease in the quality of the watermarked image is influenced by the higher  $\alpha$  value but the better level of accuracy in the extraction process [13]. In this study, the value of  $\alpha$  used is

0,1, which has been proposed in previous studies with a PSNR value of  $\geq 40$  dB [2]. After all the watermark bits are embedded, all of the DCT coefficients in colour components (RGB) are returned to the spatial domain by Equation (5).

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cdot \cos \left[ \frac{(2x+1)u\pi}{16} \right] \cdot \cos \left[ \frac{(2y+1)v\pi}{16} \right] \quad (5)$$

$x = 0,1,2, \dots, 7$  and  $y = 0,1,2, \dots, 7$ , where the magnitude of  $C(k)$  is  $\frac{1}{\sqrt{2}}$  for  $k = 0$  and 1 for the others.  $F(u, v)$  is the value of the DCT coefficient that has been embedded with a watermark bit and  $\pi = 180^\circ$ . Next, all  $8 \times 8$  DCT blocks are combined into the original image size, and the value of colour intensity in all colour components (RGB) is added by 128 to produce a watermarked image.

## 2.2. Watermark Extraction

The proposed watermark extraction is shown in Figure 2.

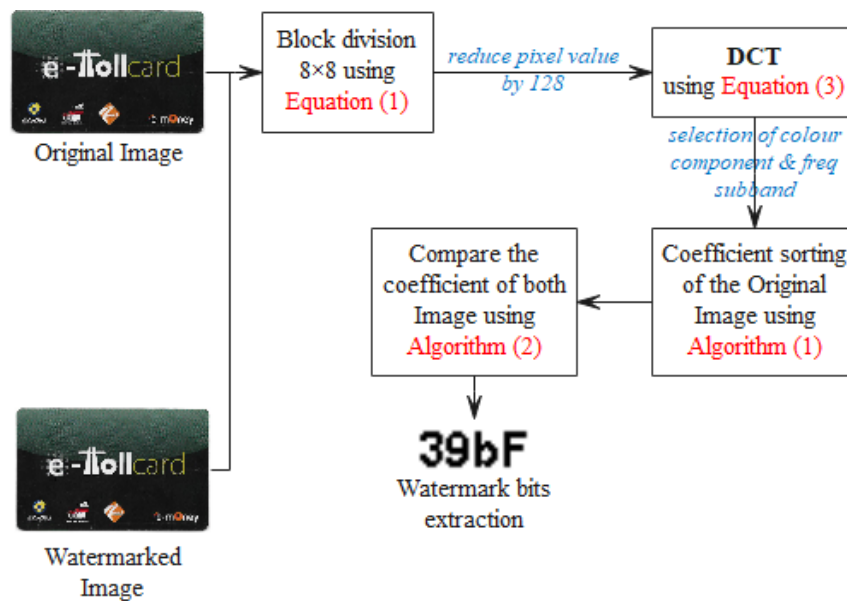


Figure 2. The proposed watermark extraction

The original image is required for the extraction process which is useful for proving ownership. To prevent watermarked images from being given a new watermark or manipulation of digital image attacks by a second party and claiming it as their own, the original owner can use the original image to prove ownership through watermark insertion and detection.

The original image and the watermarked image are divided into  $8 \times 8$  blocks, and then the value of the colour intensity (RGB) of the two images is subtracted by 128. After that, every  $8 \times 8$  block of the two images in all colour components (RGB) subtracted by 128 is transformed to the frequency domain with Equation (3). The watermark extraction process is carried out on the colour component (RGB) and the same frequency subband as when embedding.

The next step is sorting the AC DCT coefficient given in Algorithm (1), which is only done on the original image so that the pixel coordinates are the same as the watermark embedding process in each  $8 \times 8$  block. Then, the AC DCT coefficient of the original image at the pixel coordinates selected from the sorting results is compared with the AC DCT coefficient of the watermarked image at the same pixel coordinate given in Algorithm (2).

**Algorithm 2.** Extraction of watermark bits

```

1  INPUT      : The selected pixel coordinates from the sorting results are the AC DCT
                coefficient values
2  OUTPUT     : Extracted watermark bits
3  SET binary to zero
4  IF frequency ← “low” || frequency ← “middle” || frequency ← “high”
5     IF transform_coordinate_watermarked[position_x],[position_y],layer] >
   transform_coordinate_original[position_x],[position_y],layer] THEN
6     binary ← one
7     ELSE binary ← zero
8     END IF
9  END IF
    
```

If the AC DCT coefficient of the watermarked image > the AC DCT coefficient of the original image, then the watermark bit extraction result = 1, otherwise if the AC DCT coefficient of the watermarked image < AC DCT coefficient of the original image, then the watermark bit extraction result = 0, carried out until the bit extraction result watermark as the size of the original watermark image. To be displayed in the image, the value of 1 is changed to 255, and the value of 0 will still be 0.

**3. Result and Discussion**

The test was carried out on four scanned images with the HP DeskJet Ink Advantages 2135 printer with a resolution of 100 DPI in BMP format, shown in Table 2. The watermark image in the form of a combination of letters and numbers in Arial font is shown in Table 3. The programming language used for embedding analysis and watermark extraction from the proposed method is C# (SharpDevelop 5.1) with device specifications Intel (R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz, 8,00 GB RAM, 64-bit operating system, x64-based processor and Windows 10 Home Single Language.

Table 2. The original image used in the test





File Name	Original Image	Image Dimensions	File Size (KB)
flazz		338×213	211
etoll		338×216	214
matahari		338×216	214
brizzi		337×214	211

Table 3. The watermark image used in the test

Watermark	Image Dimensions	File Size (bytes)
<b>39bF</b>	35×20	224

### 3.1. Imperceptibility Analysis

A good watermark is invisible to the human eye, so it cannot distinguish between the original and the watermarked images. Reference [14] stated that the image watermarking technique aims at high imperceptibility of the embedded watermark with minimal distortion in the watermarked image. In imperceptibility, the original image and the watermarked image will be evaluated using two image quality assessment metrics commonly used in previous studies [7], [12], [15], [16], [17], namely, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

Peak Signal to Noise Ratio (PSNR) is a classic metric to estimate the similarity between two images [18]. Typical values for PSNR are between 30 and 50 dB for 8bit depth, the higher the better. PSNR qualification for the invisible watermarked image is above 45 dB on average [14]. PSNR is measured in decibels (dB) which can be obtained by Equation (6).

$$PSNR = 20 \log_{10} \left( \frac{255}{\sqrt{MSE}} \right) \tag{6}$$

Where 255 is the maximum pixel value for an 8-bit image depth, the amount of noise affecting the measured signal is represented by the MSE. Mean Square Error (MSE) is a measure used to assess how well a method is in reconstructing or restoring an image relative to the original image. The smaller the MSE value, it indicates that the image processing results are getting better or the image after processing is getting closer to the original image [19]. MSE can be obtained by Equation (7).

$$MSE = \frac{1}{M \times N} \sum_x^M \sum_y^N [f_1(x, y) - f_2(x, y)]^2 \tag{7}$$

Where  $M$  and  $N$  are the lengths of the rows and columns of the two images (original image and watermarked image),  $f_1(x, y)$  is the intensity of the image before processing, and  $f_2(x, y)$  is the intensity of the image after processing.

The Structural Similarity Index Measure (SSIM) was designed to improve traditional methods, such as PSNR, which are inconsistent with human visual perception [20]. The main emphasis in the latest research is more focused on analysing features of Human Visual System (HVS). Knowledge of combining HVS and human perception in an objective assessment of image quality can increase its accuracy [19]. SSIM measures similarity using a combination of three heuristic factors, namely the ratio of luminance, contrast, and structure [12]. The SSIM value can be obtained by Equation (8).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{8}$$

Where  $\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$  is the average intensity of the image before processing,  $\mu_y = \frac{1}{N} \sum_{i=1}^N y_i$  is the average intensity of the image after processing,  $\sigma_x^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_x)^2$  is the variance of the image before processing,  $\sigma_y^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \mu_y)^2$  is the variance of the image after processing,  $\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$  is the covariance of both images,  $C_1 = (k_1L)^2$ ;  $C_2 = (k_2L)^2$ ,  $L = 255$  if the image depth is 8 bits,  $k_1 = 0,01$ ;  $k_2 = 0,03$  is the default value of the scalar constant,  $x$  is the colour intensity value in the image before processing,  $y$  is the colour intensity value in the image after processing, and  $N$  is the number of pixels. SSIM values range from 0 to 1. If two images are similar, then SSIM is worth 1 [20].

The results of the PSNR and SSIM values for all test images are given in Table 4. The proposed method for watermark embedding produces a PSNR value of  $\geq 54$  dB with an average SSIM value of  $\geq 0,9$ . Comparison of imperceptibility for watermark embedding in each colour component (RGB) and frequency subband given the average PSNR value in Figure 3.

Table 4. MSE, PSNR, and SSIM values of the proposed watermark embedding

Frequency Subband	Image	PSNR (dB)			SSIM		
		R	G	B	R	G	B
Low	flazz	54,7350	55,0522	55,0347	0,99	0,99	0,99
	etoll	55,2678	55,1852	55,2207	0,99	0,99	0,99
	matahari	55,6545	55,0292	54,9914	0,99	0,99	0,99
	brizzi	55,0773	54,7211	54,2931	0,99	0,99	0,99
Middle	flazz	54,7342	54,9996	55,0793	0,99	0,99	0,99
	etoll	55,3730	55,2981	55,3245	0,99	0,99	0,99
	matahari	55,7737	55,0264	55,0096	0,99	0,99	0,99
	brizzi	55,0625	54,6887	54,3008	0,99	0,99	0,99

Table 4. MSE, PSNR, and SSIM values of the proposed watermark embedding (continued)

High	flazz	54,7373	55,0509	55,0322	0,99	0,99	0,99
	etoll	55,2763	55,1944	55,2306	0,99	0,99	0,99
	matahari	55,6666	55,0260	54,9896	0,99	0,99	0,99
	brizzi	55,0775	54,7162	54,3010	0,99	0,99	0,99

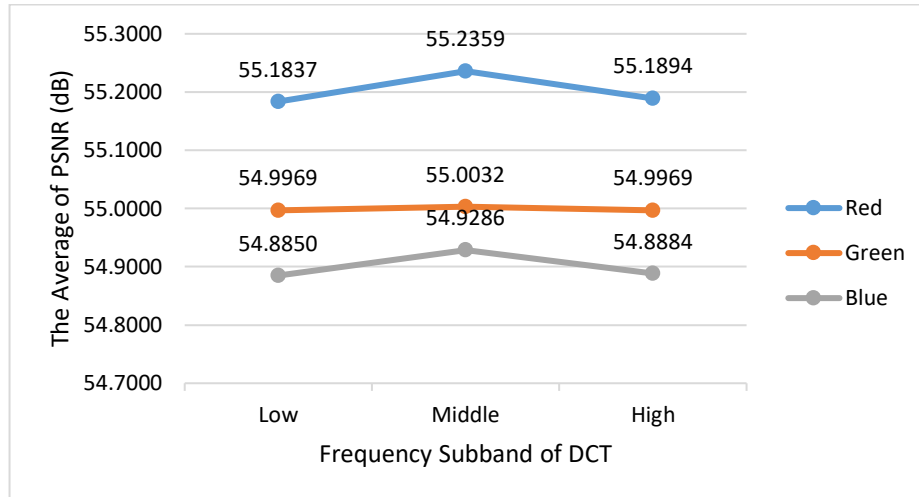


Figure 3. Graph of average PSNR for watermark embedding

This section involves describing the results obtained from the research and drawing similarities and differences between the research and previous others from methods, data, and results. However, describe whether the problems have been researched successfully according to the objectives using the proposed methods. This should involve the description of the analysis conducted, cause and benchmark of success/failure, and the unfinished part of the research followed with the steps to be taken as follow up process.

In Figure 3, it can be seen that embedding a watermark for the middle frequency in the red component produces high imperceptibility with an average PSNR value of  $\geq 55$  dB. In contrast, the low frequency in the blue component is lower than the high frequency in the green component. There is an effect on imperceptibility in selecting colour components to embed the watermark. The original image given in Table 2 is extracted from the value of the colour intensity in each colour component (RGB) and displayed in the form of an image so that the results show that the lighter image dominantly produces a higher PSNR value than the darker image. The lighter image has higher intensity values or grey levels than the darker image.

### 3.2. Robustness Analysis

Robustness is a condition in the embedded watermark that can be detected again after some common signal processing manipulation operations [21]. To analyze the robustness performance, the watermark extracted under attack was evaluated by Normalized Cross Correlation (NCC) and Bit Error Rate (BER) between the original watermark and the extracted watermark [12], [22]. Several previous studies have also used these two metrics to measure watermarking resistance [14], [16], [17]. The correlation function can determine the closeness between two digital images [19]. NCC value ranges from 0 to 1. The extracted watermark is similar to the embedded watermark when the NCC value is close to 1. The average NCC value is 1 in the case of no distortion and pixelation distortion attacks [23]. NCC can be calculated by Equation (9).

$$NCC = \frac{\sum_{x=1}^N \sum_{y=1}^M [f_1(x,y) \cdot f_2(x,y)]}{\sqrt{(\sum_x^N \sum_y^M [f_1(x,y)]^2)(\sum_x^N \sum_y^M [f_2(x,y)]^2)}} \tag{9}$$

$f_1(x,y)$  is the image intensity value before processing, and  $f_2(x,y)$  is the image intensity value after processing. Bit Error Rate (BER) calculates the bit error rate between the original and extracted watermark [12]. The lowest BER value indicates less distortion from watermark recovery [14]. The opposite of the NCC, if the value is 1, the watermark extraction results are not similar or destroyed. BER can be calculated by Equation (10).

$$BER = \sum_{i=1}^m \sum_{j=1}^n \frac{W_{i,j} \oplus W'_{i,j}}{(m \times n)} \tag{10}$$



Where  $W_{i,j}$  and  $W'_{i,j}$  are the original and extracted watermark bits with size  $(m \times n)$ , and  $\oplus$  refers to the XOR operation [12].

The results show the average value of NCC 1 and BER 0 for all original images that were watermarked without attack. The extracted watermark in the colour component (RGB) and the same frequency subband when embedding was similar to the original watermark. For robustness evaluation, watermarked images are subjected to attacks in the form of brightness +20 and contrast +50. The results of the NCC and BER values with attacks of brightness +20 are given in Table 5 for low frequency, Table 6 for middle frequency, and Table 7 for high frequency. The highest NCC value at low frequency was 0,9580 with a BER of 0,0614, and the lowest NCC was 0,7476 with a BER of 0,3429. The highest NCC value in the middle frequency is 0,9758 with a BER of 0,0357, and the lowest NCC is 0,7995 with a BER of 0,2771. The highest NCC value at high frequency is 0,9875 with a BER of 0,0186, and the lowest NCC is 0,9220 with a BER of 0,1129.

Table 5. The results of the extracted watermark image, NCC, and BER at low frequency with attacks of brightness +20













Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
Low	flazz			
		NCC=0,7503 BER=0,3371	NCC=0,7476 BER=0,3429	NCC=0,7534 BER=0,3343
	etoll			
		NCC=0,9358 BER=0,0929	NCC=0,9580 BER=0,0614	NCC=0,9521 BER=0,0700
matahari				
	NCC=0,8691 BER=0,1843	NCC=0,9281 BER=0,1043	NCC=0,9200 BER=0,1157	
brizzi				
	NCC=0,8251 BER=0,2471	NCC=0,8626 BER=0,1957	NCC=0,8428 BER=0,2229	

Table 6. The results of the extracted watermark image, NCC, and BER at middle frequency with attacks of brightness +20










Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
Middle	flazz			
		NCC=0,7995 BER=0,2771	NCC=0,9473 BER=0,0771	NCC=0,9344 BER=0,0957
	etoll			
NCC=0,9522 BER=0,0700		NCC=0,9482 BER=0,0757	NCC=0,9441 BER=0,0814	
matahari				

Table 6. The results of the extracted watermark image, NCC, and BER at middle frequency with attacks of brightness +20 (continued)

	NCC=0,9268	NCC=0,9758	NCC=0,9650
	BER=0,1057	BER=0,0357	BER=0,0514
brizzi			
	NCC=0,9095	NCC=0,9270	NCC=0,9184
	BER=0,1300	BER=0,1057	BER=0,1186

Table 7. The results of the extracted watermark image, NCC, and BER at high frequency with attacks of brightness +20

Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
High	flazz			
		NCC=0,9220 BER=0,1129	NCC=0,9342 BER=0,0957	NCC=0,9363 BER=0,0929
	etoll			
		NCC=0,9865 BER=0,0200	NCC=0,9865 BER=0,0200	NCC=0,9768 BER=0,0343
	matahari			
		NCC=0,9777 BER=0,0329	NCC=0,9817 BER=0,0271	NCC=0,9817 BER=0,0271
	brizzi			
		NCC=0,9641 BER=0,0529	NCC=0,9865 BER=0,0200	NCC=0,9875 BER=0,0186

The results of NCC and BER values with attacks of contrast +50 are given in Table 8 for low frequency, Table 9 for middle frequency, and Table 10 for high frequency. The highest NCC value in the low frequency was 0,9420 with a BER of 0,0843, and the lowest NCC was 0,7007 with a BER of 0,3957. The highest NCC value in the middle frequency is 0,9650 with a BER of 0,0514, and the lowest NCC is 0,7560 with a BER of 0,3343. The highest NCC value at high frequency is 0,9836 with a BER of 0,0243, and the lowest NCC is 0,8519 with a BER of 0,2100.

The extracted watermark image with an NCC value of  $\leq 0,90$  looks damaged and complex to detect compared to the NCC value of  $\geq 0,98$ , which is almost identical to the original watermark. Embedding watermarks at low frequencies is vulnerable to both attacks. Brightness makes each pixel in the image brighter by shifting the output transformation to the left by the desired constant amount, in this case the constant 20 is added to the pixel value. This results in a smaller pixel value changing. The brightness of an image has a high level of grey-scale values. Likewise, images that have a higher level of contrast generally display a higher level of grey-scale. Contrast enhancement in colour (RGB) images is usually accomplished by converting the image into a colour space that has luminosity. Contrast adjustment is performed on the luminosity layer only, and then the image is converted back to the RGB colour space. Manipulating luminosity affects pixel intensity. The luminance signal is performed by combining a proportion of 30 % red, 59 % green, and 11 % blue from the colour signal. Based on light waves, red has a low energy and blue has a greater energy. In this result, the green component maintains stability from attack of brightness +20 and contrast +50, while the red component is susceptible to both attacks, which are inversely proportional to the high imperceptibility.

For every original image that is watermarked in all colour components or frequency subbands with an average PSNR value of  $\geq 55$  dB, the extracted watermark can still be detected by the human eye compared to

the average PSNR value of  $\leq 55$  dB. The comparison graph of NCC with both attacks is given in Figure 4 and Figure 5 to see the difference in watermark resistance in the colour component (RGB) and the frequency subband of the proposed method.

Table 8. The results of the extracted watermark image, NCC, and BER at low frequency with attack of contrast +50













Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
Low	flazz	 NCC=0,7157 BER=0,3786	 NCC=0,7007 BER=0,3957	 NCC=0,7198 BER=0,3729
	etoll	 NCC=0,9330 BER=0,0971	 NCC=0,9420 BER=0,0843	 NCC=0,9308 BER=0,1000
	matahari	 NCC=0,9093 BER=0,1300	 NCC=0,9162 BER=0,1214	 NCC=0,8986 BER=0,1457
	brizzi	 NCC=0,8119 BER=0,2671	 NCC=0,8333 BER=0,2357	 NCC=0,7809 BER=0,3000

Table 9. The results of the extracted watermark image, NCC, and BER at middle frequency with attack of contrast +50


















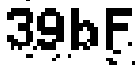






Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
Middle	flazz	 NCC=0,7560 BER=0,3343	 NCC=0,9110 BER=0,1286	 NCC=0,8870 BER=0,1629
	etoll	 NCC=0,9650 BER=0,0514	 NCC=0,9601 BER=0,0586	 NCC=0,9520 BER=0,0700
	matahari	 NCC=0,9534 BER=0,0686	 NCC=0,9591 BER=0,0600	 NCC=0,9544 BER=0,0671
	brizzi	 NCC=0,8989 BER=0,1457	 NCC=0,8862 BER=0,1643	 NCC=0,8435 BER=0,2229

Table 10. The results of the extracted watermark image, NCC, and BER at high frequency with attack of contrast +50

Frequency Subband	Image	Extracted watermark after attack		
		R	G	B
High	flazz	 NCC=0,8519 BER=0,2100	 NCC=0,8997 BER=0,1443	 NCC=0,8943 BER=0,1514
	etoll	 NCC=0,9787 BER=0,0314	 NCC=0,9787 BER=0,0314	 NCC=0,9787 BER=0,0314
	matahari	 NCC=0,9836 BER=0,0243	 NCC=0,9807 BER=0,0286	 NCC=0,9768 BER=0,0343
	brizzi	 NCC=0,9533 BER=0,0686	 NCC=0,9650 BER=0,0514	 NCC=0,9553 BER=0,0657

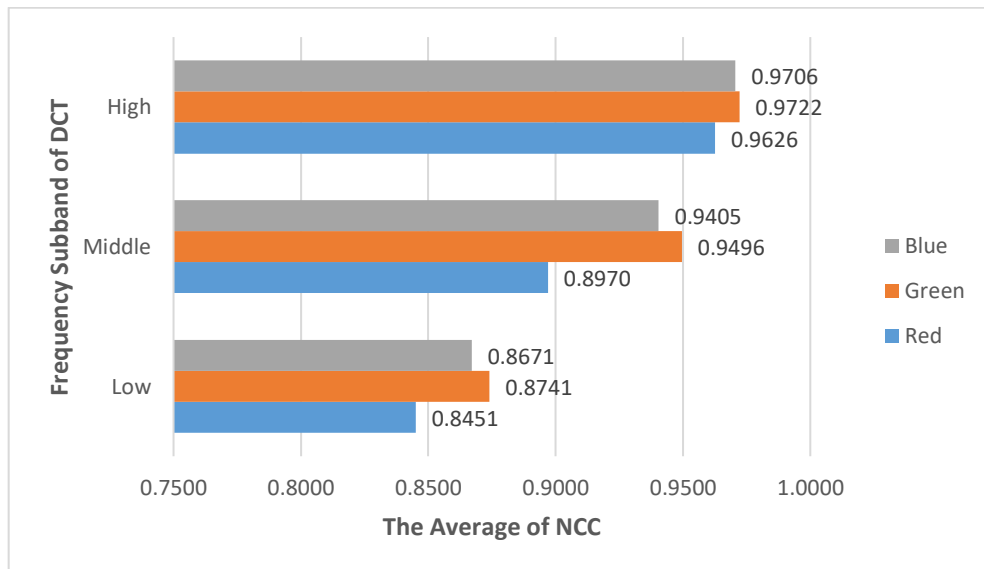


Figure 4. Graph of the average NCC with attacks of brightness +20

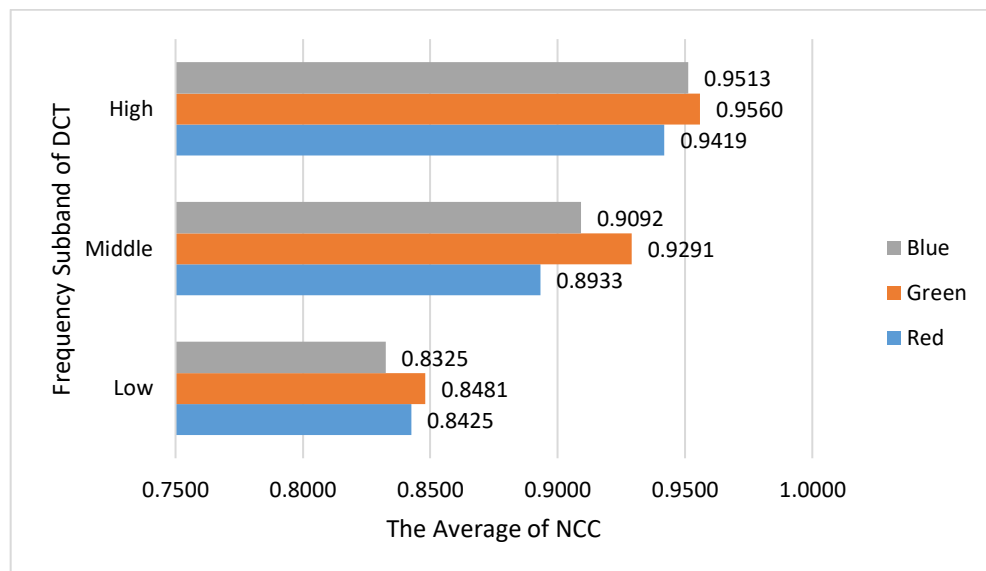


Figure 5. Graph of the average NCC with attacks of contrast +50

#### 4. Conclusions and Future Research

In this study, the embedding technique was carried out on each colour component (RGB) and Discrete Cosine Transform (DCT) frequency subband. Each  $8 \times 8$  block is embedded by a 1-bit watermark with a different embedding location. Before the watermark bit is embedded, the AC coefficient is sorted for the selected frequency subband. Each frequency subband represents the selected AC waveform and coefficient based on the cosine graph generated from the transformation. The analysis was carried out to meet the watermarking imperceptibility and robustness criteria. The proposed scheme is simple but gives results that can be knowledgeable in selecting suitable colour components and frequency subband for embedding. The results show that the green colour components and middle frequency balance imperceptibility and robustness. The red colour component produces good imperceptibility but not robustness, while the low frequency does not meet the two watermarking criteria. The proposed method successfully recovered the embedded watermark with the highest NCC of  $\geq 0,98$ . In future work, the watermark extracted from this research can be developed for edge detection to the extracted embedded information in numbers and letters to build identity card authentication applications.

#### References

- [1] C.-C. Chang, T.-C. Lu, Z.-H. Zhu, and H. Tian, "An Effective Authentication Scheme Using DCT for Mobile Devices," *Symmetry (Basel)*, vol. 10, no. 1, p. 13, 2018, doi: 10.3390/sym10010013.
- [2] N. Hasan, M. S. Islam, W. Chen, M. A. Kabir, and S. Al-Ahmadi, "Encryption Based Image Watermarking Algorithm in 2DWT-DCT Domains," *Sensors*, vol. 21, no. 16, p. 5540, 2021, doi: 10.3390/s21165540.
- [3] K. Jagdale, K. Pujari, M. Dudhanikar, and D. Gaikwad, "Digital Image Watermarking by Using Discrete Cosine Transform Methodology," *Int. J. Futur. Gener. Commun. Netw.*, vol. 13, no. 2s, pp. 521–526, 2020.
- [4] R. A. Asmara, R. Agustina, and Hidayatulloh, "Comparison of Discrete Cosine Transforms (DCT), Discrete Fourier Transforms (DFT), and Discrete Wavelet Transforms (DWT) in Digital Image Watermarking," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 2, pp. 245–249, 2017, doi: 10.14569/IJACSA.2017.080232.
- [5] R. Thanki, S. Borra, V. Dwivedi, and K. Borisagar, "An efficient medical image watermarking scheme based on FDCuT–DCT," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 4, pp. 1366–1379, 2017, doi: 10.1016/j.jestch.2017.06.001.
- [6] D. E. Kurniawan, N. R. Hartadi, and P. Prasetyawan, "Analisis Hasil Teknik Penyembunyian Hak Cipta Menggunakan Transformasi DCT dan RSPPMC pada Jejaring Sosial," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 3, pp. 261–268, 2018, doi: 10.25126/jtiik.201853692.
- [7] L. Rakhmawati, S. Suwadi, and W. Wirawan, "Blind Robust and Self-Embedding Fragile Image Watermarking for Image Authentication and Copyright Protection with Recovery Capability," *Int. J.*

- Intell. Eng. Syst.*, vol. 13, no. 5, pp. 197–210, 2020, doi: 10.22266/ijies2020.1031.18.
- [8] E. H. Rachmawanto, D. R. I. M. Setiadi, C. A. Sari, and N. Rijati, “Imperceptible and secure image watermarking using DCT and random spread technique,” *TELKOMNIKA*, vol. 17, no. 4, pp. 1750–1757, 2019, doi: 10.12928/TELKOMNIKA.v17i4.9227.
- [9] R. Krasnala, A. B. Purba, and U. T. Lenggana, “Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform ( DCT ) dan Algoritma Huffman,” *JOIN*, vol. 2, no. 1, pp. 1–9, 2017.
- [10] Z. Yuan, Q. Su, D. Liu, X. Zhang, and T. Yao, “Fast and robust image watermarking method in the spatial domain,” *IET Image Process.*, vol. 14, no. 5, pp. 3829–3838, 2020, doi: 10.1049/iet-ipr.2019.1740.
- [11] S. E. Tsai and S. M. Yang, “A Fast DCT Algorithm for Watermarking in Digital Signal Processor,” *Math. Probl. Eng.*, vol. 2017, no. Article ID 7401845, p. 7, 2017, doi: 10.1155/2017/7401845.
- [12] M. Hamidi, M. El Haziti, H. Cherifi, and M. El Hassouni, “A Hybrid Robust Image Watermarking Method Based on DWT-DCT and SIFT for Copyright Protection,” *J. Imaging*, vol. 7, no. 10, p. 218, 2021, doi: 10.3390/jimaging7100218.
- [13] P. Bogacki and A. Dziech, “Analysis of New Orthogonal Transforms for Digital Watermarking,” *Sensors*, vol. 22, no. 7, p. 2628, 2022, doi: 10.3390/s22072628.
- [14] D. Ariatmanto and F. Ernawan, “Adaptive scaling factors based on the impact of selected DCT coefficients for image watermarking,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 605–614, 2022, doi: 10.1016/j.jksuci.2020.02.005.
- [15] G. Melendez-Melendez and R. Cumplido, “Reversible image authentication scheme with blind content reconstruction based on compressed sensing,” *Eng. Sci. Technol. an Int. J.*, vol. 34, p. 101080, 2022, doi: 10.1016/j.jestch.2021.101080.
- [16] N. A. Loan, N. N. Hurrah, S. A. Parah, J. W. Lee, J. A. Sheikh, and G. M. Bhat, “Secure and Robust Digital Image Watermarking Using Coefficient Differencing and Chaotic Encryption,” *IEEE Access*, vol. 6, pp. 19876–19897, 2018, doi: 10.1109/ACCESS.2018.2808172.
- [17] L. Novamizanti, I. Wahidah, and N. P. D. P. Wardana, “A Robust Medical Images Watermarking Using FDCuT-DCT-SVD,” *Int. J. Intell. Eng. Syst.*, vol. 13, no. 6, pp. 266–278, 2020, doi: 10.22266/ijies2020.1231.24.
- [18] S.-W. Byun, H.-S. Son, and S.-P. Lee, “Fast and Robust Watermarking Method Based on DCT Specific Location,” *IEEE Access*, vol. 7, pp. 100706–100718, 2019, doi: 10.1109/ACCESS.2019.2931039.
- [19] P. N. Andono, T. Sutojo, and Muljono, *Pengolahan Citra Digital*, 1st ed. Yogyakarta: ANDI, 2017.
- [20] J. L. D. Shivani and R. K. Senapati, “Robust Image Embedded Watermarking Using DCT and Listless SPIHT,” *Futur. Internet*, vol. 9, no. 3, p. 33, 2017, doi: 10.3390/fi9030033.
- [21] M. Begum and M. S. Uddin, “Digital Image Watermarking Techniques: A Review,” *Information*, vol. 11, no. 2, p. 110, 2020, doi: 10.3390/info11020110.
- [22] F. Ernawan, D. Ariatmanto, and A. Firdaus, “An Improved Image Watermarking by Modifying Selected DWT-DCT Coefficients,” *IEEE Access*, vol. 9, pp. 45474–45485, 2021, doi: 10.1109/ACCESS.2021.3067245.
- [23] M. Ali, C. W. Ahn, M. Pant, S. Kumar, M. K. Singh, and D. Saini, “An Optimized Digital Watermarking Scheme Based on Invariant DC Coefficients in Spatial Domain,” *Electronics*, vol. 9, no. 9, p. 1428, 2020, doi: 10.3390/electronics9091428.