Invisible Colour Image Watermarking Technique for Colour Images Using DWT and SVD

Tharindu Ketipearachchi[#1], Manjusri Wickramasinghe[#2]

Abstract— **Recent popularity of social media and content sharing networks paired with fast Internet access has paved the way for mass multimedia content sharing on the Internet. As a result, volumes of sensitive data are been shared in various digital forms such that the risk of copyright infringements and misappropriations of digital content has grown at the same rate. In order to preserve copyright or authenticate digital content, digital watermarking has been considered as a viable solution for these problems. In the domain of digital watermarking, embedding a colour image as an invisible watermark to a colour cover image remain a challenge. As a solution to this problem, this paper presents such an invisible watermarking scheme based on DWT and SVD. The watermark created using the novel approach is experimentally shown to have the robustness, imperceptibility and capacity requirements.**

Keywords— **Image watermarking, DWT, SVD, Colour image watermarking, Invisible watermarking.**

I. Introduction

At present, most individuals on the Internet are used to post images and videos of their day-to-day activities on social media and sharing platforms. As a result, volumes of sensitive data have been spread across the Internet making such data vulnerable to unauthorized access. Such unauthorized access paves the way for intellectual property violations which are aggravated by the accessibility of the Internet. In recent years, unauthorized use, misappropriating, and misrepresentation [1] of digital data is on the rise with the recent technological advancements with copyright infringements and misappropriations are being considered the major threats. Due to the vast amount of data being shared, difficulty in tracking such data copyright violations are a common sight in the Internet. Furthermore, as a result such infringements, thousands of cases are being filed at courts for proof of ownership. The ability to prove the ownership of an image or digital media is the key to claim of copyright violation. However, there is no accepted standard to prove the ownership of digital data if it was not filled in an intellectual property registry which costs a significant amount of money for the digital content developers. At present, the concept of digital watermarking has emerged as a popular method to prove ownership.

When including a watermark into digital data there are a set of aspects that govern the quality of the embedding a watermark to a digital data. These aspects are that the embedded watermark should not cause damage to the actual data, be robust against possible attacks and should have a reasonable capacity. The algorithm explained in the paper represents an invisible colour image watermarking scheme with an informed (non-blind) detector based on Discrete Wavelet Transformation (DWT) and Singular Value Decomposition (SVD) for colour images. The watermarks embedded by the algorithm is experimentally shown to satisfy the quality aspects mentioned above.

The remainder of the paper is structured as follows. Section II provides the related work and the primary motivation for this research followed by the proposed method in Section III. Section IV presents the experiments conducted and the evaluation and this paper concludes in Section V.

II. Motivation

When considering the domain of digital watermarking all algorithms can be broadly categorised in two types as visible watermarking algorithms and invisible watermarking algorithms.

A. Visible Image Watermarking

In early days, most of the research on digital watermarking concentrated on visible image watermarking techniques [2] – [3]. Thus, this branch of watermarking has been studied extensively and is considered a solved problem with many algorithms meeting the necessary quality aspects of a watermark. However, the visible image watermarking is not a viable solution for the copyright infringement problem. Since the watermark is visible it can be easily cropped or digitally process to obscure such a watermark. Furthermore, such a watermark also harms the fidelity of the cover work as it deprives the user of its original experience. Such shortcomings in visible image watermarking has paved the way for invisible image watermarking.

B. Invisible Image Watermarking

The core principle behind the invisible image watermarking is that the embedding algorithm embeds an watermark to the cover image in invisible way such that it is undetectable to the user of such digital media [4]. This method is considered a better solution as it does not disturb the fidelity cover image. Number of studies exist in the literature on implementing invisible watermarking scheme. Accordingly such schemes can be categorized as belonging to four class as spatial domain techniques, frequency domain techniques, colour histogram and colour quantization [5]. Among these classes, frequency domain techniques are the popular method and techniques such as Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) and Singular Value Decomposition (SVD) are some of the mostly used frequency domain techniques [6]. Invisible image watermarking can be further divided as schemes that embed grayscale images, strings and signals and schemes that embed colour images.

1) Invisible Watermarking of Grayscale Images, Strings and Signals:

At nascent invisible watermarking schemes embeded text strings or numbers as a watermark for grayscale cover images. For example, Swanson et al. [7] proposed an algorithm to embed pseudo-noise sequences as signature for a grayscale image. Although, this algorithm has guaranteed the robustness of the watermark, its fidelity dropped when it comes to JPEG coded version of the image. Subsequent Cox

Tharindu Ketipearachchi and Manjusri Wickramasinghe are from the University of Colombo School of Computing (UCSC), Sri Lanka. (katramesh91@gmail.com[#1], *mie*@ucsc.cmb.ac.lk[#2]). Manuscript received on 2nd Nov. 2019. Recommended on 23rd June, 2020.

et al. [8] proposed a transparent watermarking scheme using frequency domain techniques Fast Fourier Transformation (FFT) and DCT. The scheme is robust against many kinds of attacks such as image cropping and JPEG compression. According to the Cox et al. only number sequence can be encoded using this algorithm. In their research they have encoded randomly generated number sequence to the colour image and recovered it successfully. Furthermore, it is illustrated that the algorithm was able maintain the quality of the cover image after the watermarking process as well. This is comparatively better than the approach introduced in Swanson's et al. [9].

Similarly, Fleet and Heeger employed sinusoidal signal embedding as a way to achieve invisible watermarking in colour images [10]. The techniques involved using S-CIELAB technique on the Yellow and Blue components of the cover image for signal encoding. The scheme is shown to be robust, efficient and the deviation between cover and watermarked images are reasonably low. However, this scheme is only supported for sinusoidal signal encoding. Another blind watermarking scheme based on the DWT and the Arnold transformation was proposed by Dharwadkar and Amberker [12]. This algorithm has facilitated an opportunity to embed vector image as a watermark instead of the signals and strings. Although, this algorithm was illustrated to be resilient to different types attacks, the decoded watermark image is bit unclear. Santi and Thangavelu has also suggested an approach to embed vector image to a colour image [13]. They have proposed an algorithm in which the features of DWT, DCT, SVD techniques are combined. This solution is robust against various kind of attacks such as salt and pepper noise, Gaussian noise, Gaussian blur, cropping, colour contrast and compression attacks. Similarly, watermarks embedded in high frequency components are resistant to image sharpening, histogram equalization, and resize. Extracted watermark also with the reasonably better quality. But the proposed algorithm is not robust to the rotation attack. Humming et al. [14] proposed another approach to embed grayscale vector image to a colour image [14]. The algorithm is an efficient watermarking embedding algorithm for colour image which combines the spectral characteristics of HVS and the characteristics of green component in a colour image. However, this proposed algorithm is less robust against rotating. When considering the deviation between the cover image and watermarked image, quality of the extracted image, is minimal. When we consider above researches, there are number solutions available for invisible image watermarking on grayscale images as well as text string and grayscale image embedding on colour images. Hence, it can be construed to be a solved problem.

2) *Invisible Watermarking of Colour Images:*

When considering the embedding of a colour image on to a colour image cover several solutions are found in literature. However, the challenge is to achieve all the basic watermarking requirements in single approach. Imperceptibility, capacity and robustness are the major requirements of a good watermarking scheme. Finding such a scheme in this category of invisible watermarking remains a challenge. Chou and Wu [15] has proposed an efficient algorithm for invisible colour image watermarking based on the image quantization approach for watermark encoding. The scheme is computationally simple and quite robust in face of various attacks such as cropping, low-pass filtering, white-noise addition, scaling, and JPEG compression with high compression ratios. However, the algorithm is only tested for vector image and when raster colour images as watermarks are used there might be issues with extracted watermark. Bas et al. [16] suggested an invisible colour image watermarking scheme using the hyper complex numbers representation and the Quaternion Fourier Transform (QFT). Eventhough, this scheme was able to embed a watermark with minimum deviation of the cover image, they have not described much about the watermark extractions results. However, this scheme has evaluated for different colour image filtering process (JPEG, blur) and the fact that perceptive QFT embedding can offer robustness to luminance filtering techniques. Chan et al. [17] and others proposed another image quantization based watermarking scheme. This scheme used colour quantization and principal component analysis (PCA). The image quality of the extracted watermark image is of acceptable quality. However, this algorithm is not properly tested against attacks. Robustness of the algorithm yet to be assured. Mohanty et al. [18] proposed a novel approach to invisible colour image watermarking using DCT. But in here, they insert watermark image to part of the base image to make the actual watermark image (perceptually) which is very unclear. After the extraction process they were only able to obtain that perceptual image which is unclear. The size of the watermark image is very small. However, the robustness of the algorithm is assured. Kaarna et al. [19] have proposed another scheme using ICA, DWT and DCT with unclear extracted watermark. Agarwal and Venugopalan [20] has suggested a scheme that divide both cover image and watermark image to its RGB components and embed respective components separately [20]. The algorithm is based on the well-known matrix factorization technique of the singular vector decomposition. The robustness of the algorithm was proven against various attacks. But the extracted watermark image is not sufficiently clear enough. Chaitanya et al. [21] and others suggested a scheme using DWT and DCT. They have used colour vector image as a watermark. Performance with the raster images may be doubtful. Since only Blue component is used to embedding, size of the watermark image is small when compared to the cover image. In here they have used 1024×1024 cover image and 32×32 watermark image. Pradhan [4] has suggested an approach for invisible image watermarking using DCT, DWT and SVD. Although, in grayscale images this approach provides accurate results, the algorithm fails in coloured watermark images. It failed to extract similar image to the original watermark image from the watermarked image. Vishnavi and Subashini has proposed scheme using SVD [22]. This algorithm can embed and extract the colour image perfectly. But in this they are only embedding blue component of the watermark image to the cover image.

When considering the above solutions it can be concluded that embedding colour image to a colour image as an invisible watermark still remain an open problem. Different researchers have achieved some of the basic watermarking requirements. But none of them were able to achieve all the basic requirements together in one solution. There is no perfect solution found yet. Almost all of them are need to be improved.

III. PROPOSED METHOD

As per the background research, there are number of issues identified with respect to invisible colour image watermarking on a colour image cover. The issues identified on various schemes can be categorized into three aspects as attempting to solve the problem using single technique such as DCT or incorrect combination of techniques, applying the techniques that were successful with grayscale images to the colour images in a similar manner, trying to embedded watermark to the entire image.

By considering the pros and cons of technologies of each domain we have selected DWT and SVD as our base techniques to build our novel watermarking scheme. Frequency domain was chosen due to the higher compression ratio and good localization and DWT can preserve a higher fidelity and a higher capacity. However, as observed with other frequency domain techniques as well, DWT is less robust against attacks. In order to overcome this vulnerability, DWT was combined with SVD which makes watermarks more robust against attacks. Furthermore, SVD does not compromise the basic strengths of the DWT techniques. To achieve the desired robustness against attacks, we augment the the resulting image matrix with the SVD components without changing the original output form.

1) Discrete Wavelet Transform (DWT)

DWT is a neoteric technique consecutive used in digital image processing, compression, digital watermarking etc [6]. This is widely used very popular technique in digital watermarking domain since its more efficient than other similar techniques such as the DCT. In DWT, image is dissolved into high and low frequency elements in two level discrete wavelet transform (DWT). The robustness with respect to various attacks increases when the watermark is embedded in low frequencies gained by the WD (wavelet decomposition). At first, the digital media is segmented into frames, then discrete wavelet transform is applied to luminance element of each frame which outcomes into discrete sub bands. Subsequently these bands are dissolved into discrete components and the covariance matrix is computed for each such component. The watermarked luminance component of the frames is obtained by applying inverse discrete wavelet transform. Ultimately watermarked digital media is gained by renewing the watermarked frame [6,23,24].

2) SVD(Singular Value Decomposition):

Singular value decomposition is a numerical technique which is utilized to diagonalize matrices in numerical analysis [25]. In variety of applications singular value decomposition is used as an algorithm. In this singular value decomposition transformation, one matrix can be dissolved into three matrices. These matrices are of the equal size as the original matrix. By the linear algebra, an image is an array of nonnegative entries of scalar values that can be deduced as a matrix [23].

A. *DWT/SVD Based Approach for Invisible Image Watermarking.*

The proposed watermarking scheme uses the 2-dimensional DWT with 'haar' signal along with SVD. The major concept of suggested approach is to segment both cover and watermark colour images to its red, blue, green components and apply the watermarking process separately for each component. Subsequently, merge the watermarked components again to form the watermarked image. The DWT/SVD major image watermarking algorithm explained in Section III.2(B) followed by the component based watermarking algorithm to embed watermark image to red, blue, green components in Section III.2(C).

B. *DWT/SVD Major Image Watermarking Algorithm*

In this major watermarking algorithm both the cover image and the watermark image will be decomposed to red, blue, green components. As we illustrated in the Figure 1, cover image is the image which we are going to insert the watermark. Watermark image is the image which we are going to insert as the watermark. First, we decompose cover image into the components R, G, B components. B_r, B_g, B_b are the resulted red, green, blue components of the cover image respectively. By applying the same process to the watermark image, W_r, W_g, W_b will output as the respective red, green, blue components of the watermark image.

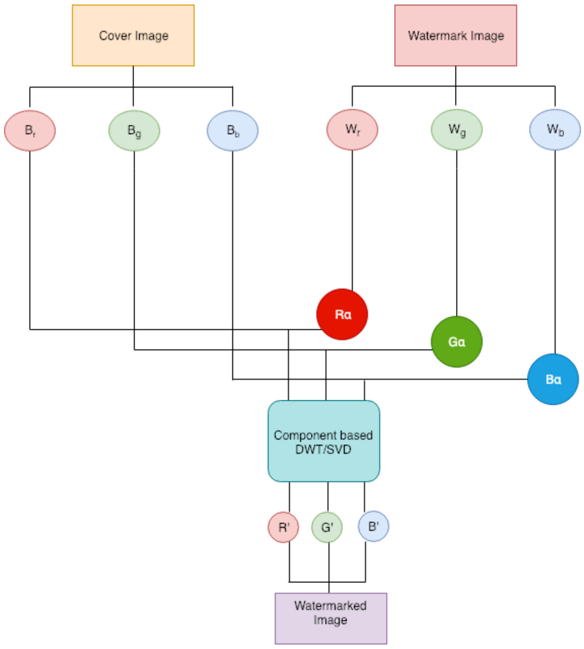


Fig. 1 DWT/SVD Major Image watermarking algorithm

As illustrated in Fig. 1, the decomposed components are then used as the inputs to the component based watermarking algorithm component based watermarking algorithm which will be explained in the next Section III.2(C). The inputs to the component-based algorithm are of three components. These are the corresponding RGB components (e.g. B_r, W_r) of the cover and the watermark image and the watermark strength factor ($0 \leq \alpha \leq 1$). Once the component-based algorithm is applied to the each components of the cover and watermark images, the algorithm will output the corresponding component of the watermarked image (e.g R'). This process is applied to all components of the images separately by providing a separate alpha value.

C. *DWT/SVD Component based Watermarking Algorithm*

As mentioned in the previous section component-based algorithm designed as suitable for 1-layer images (e.g single component of an image in case of RGB). The algorithm applies transformation techniques to the image components followed by the embedding of the watermark as detailed below.

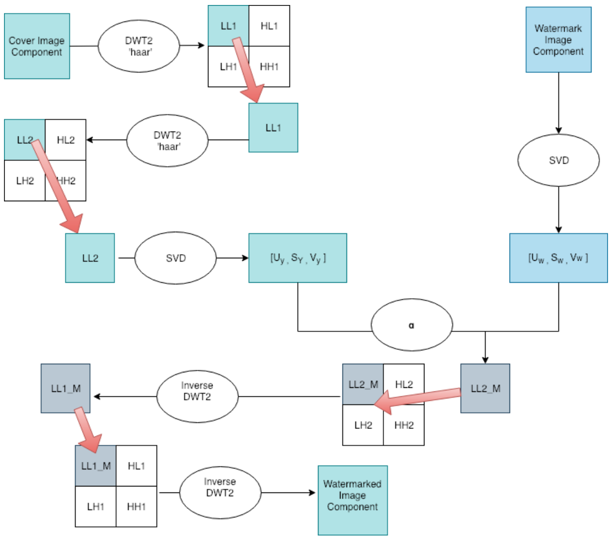


Fig. 2 DWT/SVD component based watermarking process

As illustrated in Fig. 2, initial step is to apply 2-dimensional DWT to the cover image component using 'haar' DWT signal type. As a result, the approximate image (LL1) and other detail components of the image (HL1, LH1, HH1) are obtained. In the second step the approximate image component (LL1) is reapplied with the 2D DWT transformation that results in outputs LL2, HL2, LH2, HH2 components respectively. As the third step, SVD is performed on the LL2 component and the watermark image component and obtain the three matrices resulting metrices [U$_y$, S$_y$, V$_y$] and [U$_w$, S$_w$, V$_w$] respectively. Then the S matrices of both derivations are combined with the alpha value to obtain the new S component. The combination of the S components are obtained as:

$$S_{mark} = S_y + \alpha \times S_w$$

Where, S_{mark} : combined S component, α : Embedding strength ; { $0 \leq \alpha \leq 1$ }, then we rebuild the approximate image component LL2 such that,

$$LL2_M = U_y \times S_{mark} \times V_y'$$

Where, U_y : U component of the final processed watermark output, $V_y^{'}$: Transpose matrix of V_y , $LL2_M$: New LL2 component. Then we combine this $LL2_M$ component with early derived HL2, LH2, HH2 components and apply the inverse 2D DWT process, which will derived another approximate image component ($LL1_M$).

$$[LL2_M, HL2, LH2, HH2] \frac{IDWT2}{haar} \rightarrow LL1_M$$

Then combine $LL1_M$ component with HL1, LH1, HH1 and apply inverse 2D DWT process again. Which will gives the watermarked image component (WM) as the result.

$$[\ LL1_M, HL1, LH1, HH1] \frac{IDWT2}{haar} \rightarrow WM$$

D. *DWT/SVD Major Image Watermark Extraction Algorithm.*

This algorithm is also based on red, green, blue component decomposition. A non-blind watermark extraction approach is followed in that some of the components of original watermark image is required for watermark the extraction process. However, the entire image is not required in the extraction process. The watermark extraction process is similar to the embedding process in which a Major Algorithm is followed by the component-based application. In this section we'll be explaining the major watermark extraction algorithm.

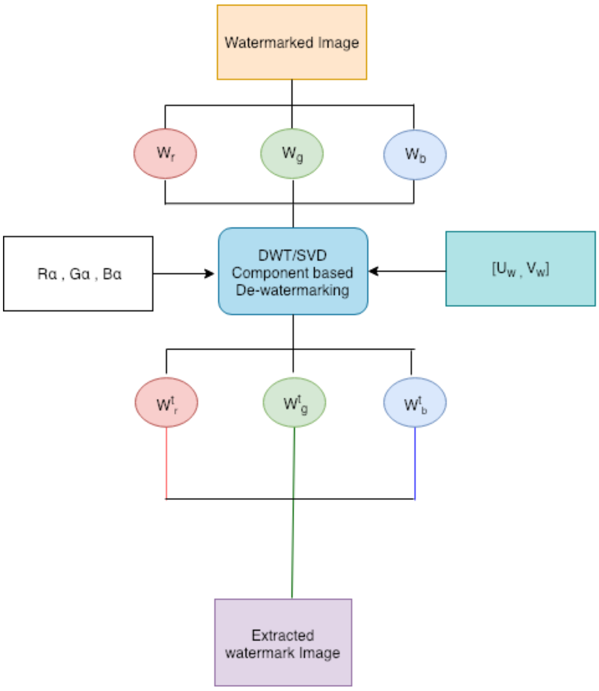


Fig. 3 DWT/SVD Major image watermark extraction algorithm

As illustrated in Fig. 3, the first step is to decompose watermarked image to the red, blue, green components. Then the relevant alpha value along with the $[U_w , V_w]$ components of the watermarked image and apply the component-based watermark extraction process to each component separately. After providing required inputs, the component-based algorithm will generate the extracted watermark image component. By merging the three red, blue and green components extracted watermark image will be formed.

E. *DWT/SVD Component Based Watermark Extraction Algorithm.*

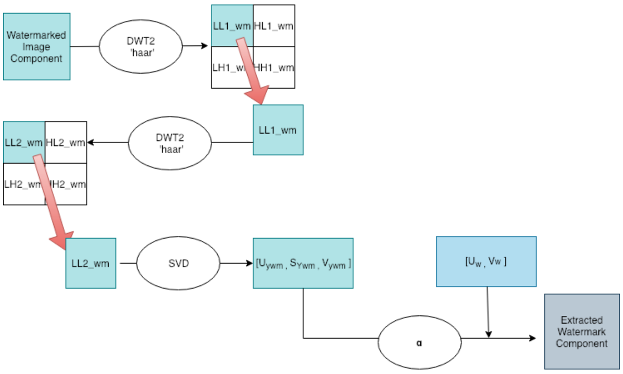


Fig. 4 DWT/SVD component-based watermark extraction process.

Fig. 4 depicts the complete flow of the component-based watermark extraction algorithm. Initially as the 2D DWT to the watermarked image component with ‘haar’ signal type is applied. This process will decompose image to approximate image and detail components [LL1wm, HL1wm, LH1wm, HH1wm]. Then take the LL1wm component and reapply 2D DWT that results in another four components [LL2wm, HL2wm, LH2wm, HH2wm]. Now we take the generated LL2wm component and apply SVD to generate three independent matrices [U_{ywm} ,S_{ywm} ,V_{ywm}] as the output. Using S component of this matrix and S component of the previously SVD decomposed matrix of the cover image (S_y) we’ll generate the S component of extracted image (S_{wrec}).

$$S_{wrec} = \left(S_{yw} - S_y\right) / \alpha$$

Using this S_{wrec} component and provided [U_w, V_w] components of the original watermark image, watermark image component will be generated.

$$IMG_{ex} = U_w \times V_w$$

In here we have used SVD theory and product of the U_w , S_{wrec} matrices, along with the transpose of the matrix V_w extracted watermark component will be formed.

IV. Experiment and Evaluation

There are three main conflicting properties of watermarking algorithm which have been used to measure the quality of a good watermarking algorithm. In the experiment setup we attempt to prove that the proposed approach has achieved these conflicting requirements to an acceptable level. Those conflicting properties of good watermark are imperceptibility, robustness and capacity. Apart from the main requirements of the algorithm there is another additional requirement. That is watermark image and extracted watermark should look similar or recognizable. In other words, the deviation between watermark image and the extracted watermark also should stay minimum. Separate experiments are designed to evaluate each property and requirements. The deviation between watermark and extracted watermark is also evaluated in each experiment for conflicting properties. We will be discussing the experiment as well as analyse its observations in this section.

A. *Measurements*

1) Mean Square Error *(MSE)*

The MSE measures the error with respect to the centre of the image values, i.e. the mean of the pixel values of the image, and by averaging the sum of squares of the error between the two images.

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{i=0}^{n-1}\left(p_1(i,j) - p_2(i,j)\right)^2$$

Where, p_1 (m, n) and p_2 (m, n) represent two images of size m × n. A lower value of MSE signifies a lesser error in the reconstructed image [26].

2) Peak Signal to Noise Ratio (PSNR)

The PSNR estimates the quality of the reconstructed image in comparison to the original; it is a standard way of measuring image fidelity [27]. Here ‘signal’ is the original image and ‘noise’ is the error in the modified image. PSNR is a single number that reflects the quality of the reconstructed image and is measured in decibels (dB). It is most easily defined using the mean squared error (MSE) for two monochrome images p1 and p2 where one of the images is considered a noisy approximation of the other. The PSNR is defined as:

$$PSNR = 10\log_{10}\left(\frac{MAX_{p1}^2}{MSE}\right) = 20\log_{10}\left(\frac{MAX_{p1}^2}{\sqrt{MSE}}\right)$$

Here, MAX_{p1} is the maximum pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255 [26].

B. *Imperceptibility*

Imperceptibility is also called as the fidelity. Fidelity corresponds to the visibility of artefacts introduced into an image by the watermarking process. Simply it is the “perceptual similarity” between watermarked and un-watermarked versions of an image. This is basically the difference between cover image and the watermarked image. After the watermark embedding process, still cover image should stay same as the original cover image. In order to evaluate the imperceptibility of the proposed algorithm, the difference between cover image and the watermarked image is computed once watermarking process has completed. The PSNR and MSE values between cover image and watermarked image is used to quantify the deviations between the images.

1) Experiment

In this experiment we evaluate the imperceptibility of the proposed method. In the same setup the alpha values of the three components are iteratively changed to obtain an optimized alpha values set for the suggested approach. Fig. 5 depicts the high-level overview of the experiment.

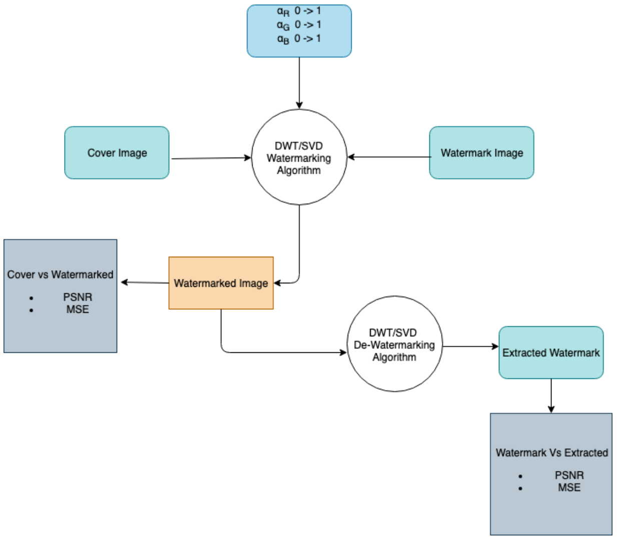


Fig. 5 Imperceptibility evaluation experiment design.

In this experiment, the alpha value is increased by 0.1 for each execution of the experiment. Initially, all possible alpha value combinations for the red, blue green alpha components are obtained. The for each alpha value combination, cover image and watermark image will be given as inputs and watermarking algorithm will be executed and PSNR and MSE values between cover image and the watermarked image will be computed and recorded. Subsequently to the embedding of the watermark, the extraction of the watermark is also performed for all the combinations and the PSNR and MSE values between the watermark image and the extracted watermark image is obtained.

2) *Results Analysis and Evaluation*

Following data have been observed from the experiment explained in Section IV.B(1) above. An excerpt of the recorded data will be shown here.

TABLE I
EXTRACTED PART OF PSNR, MSE VALUES OF OBSERVATION ACCORDING TO DIFFERENT ALPHA VALUE COMBINATIONS.

R Alpha	G Alpha	B Alpha	Without Attacks			
			Cover Vs Watermarked		Watermark Vs Extracted	
			PSNR	MSE	PSNR	MSE
0.1	0.1	0.1	37.404	11.823	28.686	88.008
0.1	0.1	0.2	34.066	25.494	28.686	88.008
0.1	0.1	0.3	31.311	48.085	28.686	88.008
0.1	0.1	0.4	29.121	79.607	28.686	88.008
0.1	0.1	0.5	27.343	119.89	28.686	88.008
0.1	0.1	0.6		168.82	28.686	88.008
0.1	0.1	0.7	24.584	226.3	28.686	88.008
0.1	0.1	0.8	23.478	291.94	28.686	88.008
0.1	0.1	0.9	22.498	365.84	28.686	88.008
0.1	0.1	1	21.621	447.72	28.686	88.008
0.1	0.2	0.1	34.355	23.854	28.686	88.008
0.1	0.2	0.2	32.388	37.525	28.686	88.008
0.1	0.2	0.3	30.341	60.116	28.686	88.008
0.1	0.2	0.4	28.51	91.638	28.686	88.008
0.1	0.2	0.5	26.928	131.93	28.686	88.008
0.1	0.2	0.6	25.558	180.85	28.686	88.008
0.1	0.2	0.7	24.359	238.33	28.686	88.008
0.1	0.2	0.8	23.302	303.97	28.686	88.008
0.1	0.2	0.9	22.357	377.87	28.686	88.008
0.1	0.2	1	21.506	459.75	28.686	88.008
0.1	0.3	0.1	31.721	43.749	28.686	88.008
0.1	0.3	0.2	30.54	57.42	28.686	88.008
0.1	0.3	0.3	29.099	80.011	28.686	88.008
0.1	0.3	0.4	27.657	111.53	28.686	88.008
0.1	0.3	0.5	26.317	151.82	28.686	88.008
0.1	0.3	0.6	25.104	200.75	28.686	88.008
0.1	0.3	0.7	24.011	258.23	28.686	88.008
0.1	0.3	0.8	23.027	323.87	28.686	88.008
0.1	0.3	0.9	22.135	397.77	28.686	88.008
0.1	0.3	1	21.322	479.65	28.686	88.008

Table 1 has shown the extracted part of the data set which we had received the above described experiment. One can notice here that the PSNR and MSE values has been changed between cover image and watermarked image. But the PSNR and MSE values between watermark image and extracted watermark remains same. It didn't change according to the alpha value combinations.

As we have explained in the Section III, the following formula was used to embed S component of watermark to S component of the cover image.

$$S_{mark} = S_y + \alpha \times S_w$$

We have rebuilt the S component as per the above formula. For that we have used alpha value. We used 0 – 1 value as an alpha value. Because above formula will be applied to the points of the image matrix of each red, blue, green component. Here each point represents the pixel value. Pixel value takes 0.0 to 255.0 value. When the value is closes to 0 it will get further darker and when the value closes to the 255.0, it will get further bright.

According to the above formula we are incrementing the value of the pixel. That means after the watermarking process, our cover image gets further brighter. But if it gets too much brighter that'll be easily noticeable to the human eye. Therefore, in order to minimize the brightness deviation as result of the embedding process alpha is chosen. So, a value between 0 and 1 is used as an alpha value, because pixel value cannot take a higher value than 255. So, we multiply the pixel value of watermark component by decimal value and try to keep the value of rebuild component less than 255. Since we have applied SVD to the image component and get the one matrix of the output there will not be any values close to the 255, because SVD has decomposed single image matrix to product of 3 matrices. When we take 0.1 as an alpha value, it can be observed that the value of each pixel changes between 1.0 to 2.0 intensity levels. That means watermarked image get brighter but it's not a noticeable amount. Then we used 1.0 as the alpha value, you can see that the value of the pixel is changed 15.0 to 16.0 intensity level which is a perceptible change. Values of alpha greater than 1.0 results in a perceptible change in the intensity of the pixel which is easily observed.

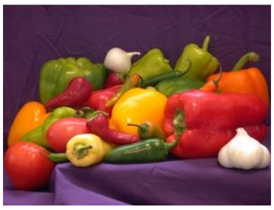


Fig. 6 Original cover image.

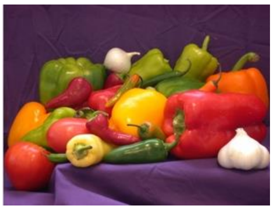


Fig. 7 Watermarked image

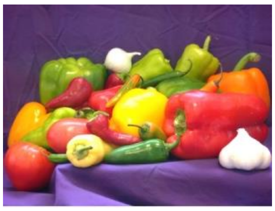


Fig. 8 Watermarked image using 1.0 alpha value.

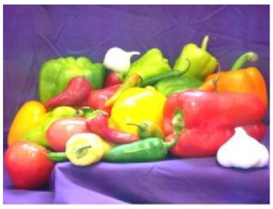


Fig. 9 Watermarked image for 2.0 alpha value.

The Figs 6, 7, 8 and 9 has depicts the original cover image watermarked image 0.1, 1.0 and 2.0 alpha respectively. Here it's obvious that original cover image and 0.1 alpha watermarked image looked same. There is no perceptually noticeable difference between images. But when it comes to 1.0 alpha, image get much brighter and difference is noticeable. But when it comes to 2.0 alpha image has lost his identity as well. Due to the reasons stated above, the alpha value range between 0 and 1.0 was considered with 0.1 increments.

The following Figure 10 has shown the original image which we used as a watermark. Figure 11, 12, 13 showed the extracted watermark image for watermark embedding using 0.1, 1.0, 2.0 alpha values respectively. Here you can see the extracted watermark look same for every value. That doesn't simply change according to the alpha value. That's because the following formula was utilized to extract the S component.

$$S_{wrec} = (S_{yw} - S_y) / \alpha$$

Simply alpha value is eliminated from the total calculation process. Hence, the extracted image is independent of alpha values unless pixel values has been changed by an attack or other kind of processing.

Now we move on to the recorded data by the algorithm. The image with the minimum MSE and maximum PSNR is said to be the most successful image. Because minimum MSE and maximum PSNR means the minimum deviation of the image from the original image. By looking at the maximum PSNR we could find the optimal combinations of the alpha values. Table II has shown the extracted part of the recorded data. It shows that maximum PSNR is 37.404 and minimum MSE is 11.823. Both are given in to the same alpha value combination which is 0.1 alpha for all components. If PSNR is higher MSE will get lower because PSNR is directly calculated using MSE. According to the formula which we have discussed earlier. Because of that considering one variable of them is enough for the analysing process.

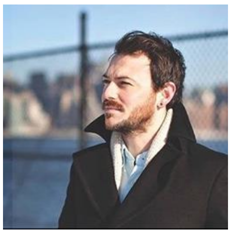


Fig. 10 Original watermark image.

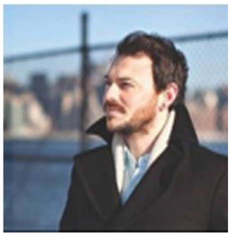


Fig. 11 Extracted watermark image for 0.1 alpha embedding.

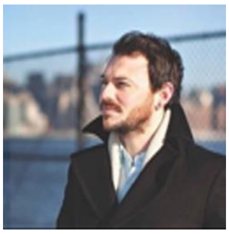


Fig. 12 Extracted watermark image for 1.0 alpha value.

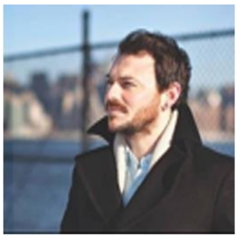


Fig. 13 Extracted watermark image for 2.0 alpha value.

TABLE II
OPTIMIZED ALPHA VALUE COMBINATION FOR WATERMARKING WITHOUT ATTACKS

R Alpha	G Alpha	B Alpha	Without Attacks			
			Cover Vs Watermarked		Watermark Vs Extracted	
			PSNR	MSE	PSNR	MSE
0.1	0.1	0.1	37.404	11.823	28.686	88.008
0.1	0.1	0.2	34.066	25.494	28.686	88.008
0.1	0.1	0.3	31.311	48.085	28.686	88.008
0.1	0.1	0.4	29.121	79.607	28.686	88.008
0.1	0.1	0.5	27.343	119.89	28.686	88.008
0.1	0.1	0.6	25.856	168.82	28.686	88.008
0.1	0.1	0.7	24.584	226.3	28.686	88.008
0.1	0.1	0.8	23.478	291.94	28.686	88.008
0.1	0.1	0.9	22.498	365.84	28.686	88.008
0.1	0.1	1	21.621	447.72	28.686	88.008
0.1	0.2	0.1	34.355	23.854	28.686	88.008

Fig. 14 depicts the results of the watermarking process of 0.1, 0.1, 0.1 alpha value combination without attacks.

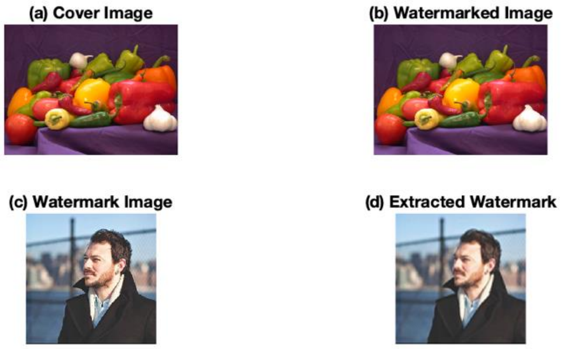


Fig. 14 Watermarking resulted images for 0.1, 0.1, 0.1 alpha value combination

C. *Robustness*

Robustness basically means the stability against different kind of attacks. There are various kinds of attacks being executed against images. A watermarking algorithm should be implemented in a way that simply attackers cannot remove the watermark by executing attacks against watermarked image. In order to analyse the robustness of suggested approach against different kind of attacks we have designed following experiment.

1) Experiment

Fig. 15 depicts the experimental design. This experiment also designed with the same way as the experiment we had discussed in imperceptibility section. Small modifications have been done to the initial experiment design. Initially we change the alpha values and do this experiment for different kind of alpha value combinations which is same as the experiment in previous section. Then after the watermarking process is complete attacks will be performed on the watermarked images and the PSNR and MSE measures between cover image and the watermarked image. Seven (07) types of attacks which is most common and effective in watermarking domain are executed to the watermarked image. Those are,

- Gaussian Noise attack.
- Salt and Pepper attack.
- Median Filter attack.
- JPEG Compression attack.
- Mean Filter attack.
- Butterworth High pass Filter attack.
- Butterworth Low pass Filter attack.

For each attack, optimized alpha value combination will be captured. Finally, we will be come up with the generalized alpha value combinations which is suitable for all the attacks and attack-less situations.

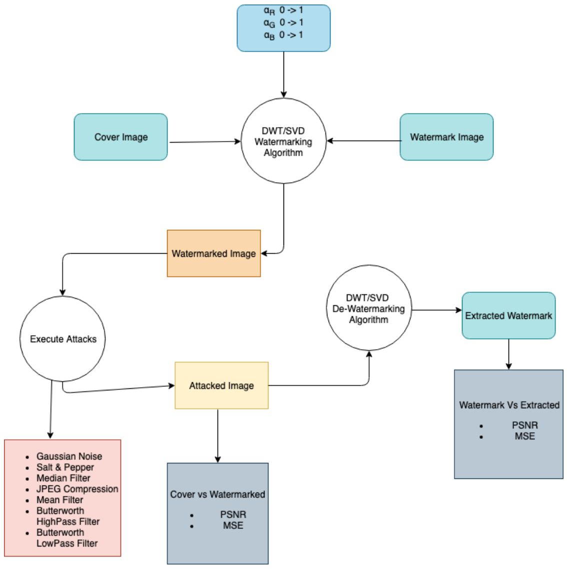


Fig. 15 Robustness evaluation experimental design.

2) Results Analysis and Evaluation

PSNR and MSE values between cover and attacked images and PSNR, MSE values between watermark and extracted watermark images have been collected for the seven attacks. Data has been recorded for 1000 alpha value combinations. Then recorded data has been analysed in a stepwise manner as explained below.

As the first step we looked at the deviation between attacked image and the watermarked image. Table III has provided the summary of maximum PSNR and minimum MSE values for each of seven attacks.

TABLE III
SUMMARY OF MINIMUM IMAGE DEVIATION BETWEEN COVER IMAGE AND ATTACKED IMAGE FOR ALL ATTACK TYPES.

Type of the Attack	Max PSNR	Min MSE	R Alpha	G Alpha	B Alpha
Without Attacks	**37.404**	**11.823**	**0.1**	**0.1**	**0.1**
Gaussian Noise	11.823	598.15	0.1	0.1	0.1
Salt & Pepper	**17.64**	**1119.5**	**0.1**	**0.1**	**0.1**
Median Filter	35.242	19.45	0.1	0.1	0.1
Mean Filter	**37.404**	**11.823**	**0.1**	**0.1**	**0.1**
JPEG Compression	29.652	70.449	0.1	0.1	0.1
Butterworth High-pass	**13.323**	**3025.5**	**0.1**	**0.1**	**0.1**
Butterworth Low-pass	10	6501.8	0.1	0.1	0.1

Here we have concluded that minimum image deviation is received for 0.1, 0.1, 0.1 alpha values for every attack. In previous experiment we have concluded that this is the optimal alpha values combination for the image watermarking without any attacks. Here what we have done was repeat the same process of the previous experiment and then execute an attack to the watermarked image, then measure the PSNR and MSE between attacked image and the cover image. According to above results, it is evident that every attack has deviated the image by some constant amount from the watermarked image. Because of that minimum deviated image in the watermarking process has the minimum deviation against the attack as well.

Hence, we could not observe a deviation between cover image and the attack image as the factor to measure the robustness of our algorithm. We have few reasons for that. First thing is robustness of the watermarked image against attacks is not only depend on the watermarking algorithm. There may other reasons for that as well. Structure of the original image, quality and nature of the original image and lot of factors of the original image has been affected to that. There are two main objectives of our algorithm, first one is to minimize the deviation between cover and the watermarked image and second one is to minimize the deviation between watermark and the extracted watermark images. In order to fulfil the objective of an efficient watermarking scheme one can remove the deviation between cover and attacked images from consideration. The primary focus should be on the deviation between watermark image and the extracted image. The robustness of the watermark against attacks means the embedded watermark image would stay same on any kind of attack. Keep extracted watermark image's deviation minimum against the attacks is the accurate factor to measure the robustness of the watermarking algorithm. Here we have found the optimal alpha value combination which made extracted watermark image's deviation minimum against each attack.

TABLE IV
OPTIMIZED ALPHA VALUE COMBINATION FOR EACH TYPE OF ATTACK.

ATTACK	MAX PSNR	MIN MSE	R Alpha	G Alpha	B Alpha
Without attack	28.686	88.008	0.1	0.1	0.1
Gaussian	15.237	1947.2	0.7	0.6	0.8
Salt & Pepper	15.382	1883.1	0.7	0.9	0.9
Median Filter	19.683	699.42	0.1	0.3	0.3
Mean Filter	17.766	1087.5	0.3	0.4	0.6
JPEG Compression	18.082	1011.4	0.2	0.4	0.3
Butterworth High-Pass	6.7039	13890	1	1	1
Butterworth Low-Pass	5.7119	17454	1	1	1

Table IV has shown the optimal alpha value combination for each attack. Here we could see that strength of the attacks has been increased from top to bottom the table. PSNR value get decreased according to the strength of the attack. Mean time we can observe that the respective alpha value combinations also get increased gradually. Fig. 16 illustrates the mean of the alpha value combinations against the PSNR value.

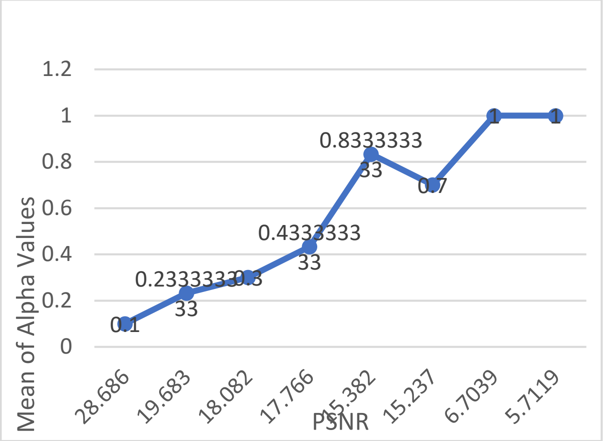


Fig. 16 Mean of alpha values variation between PSNR.

PSNR values has opposite relationship with strength of the attack. Lower the PSNR means attack is much stronger. As the graph of the Fig. 16 has depicted optimum alpha value has increased according to the increment of the strength of an attack. We can conclude that when alpha value get higher watermark would be more robust against attack. Conversly, on the other hand when alpha values get increased image deviation between cover and the watermarked image get increased which means image get much brighter. Hence a balance between the alpha values and perceptibility is required.

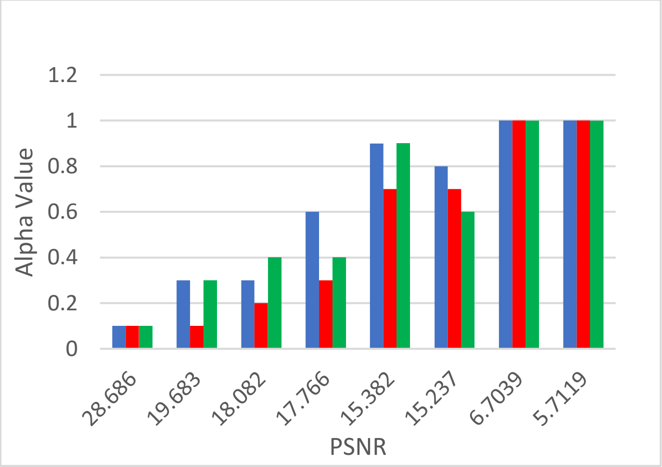
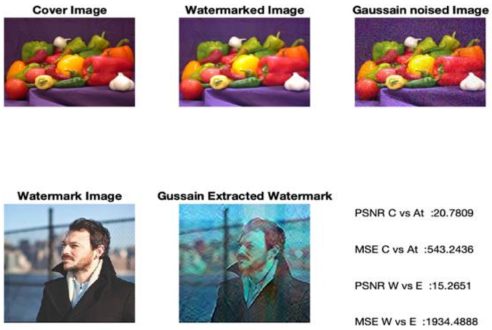


Fig. 17 Red, Blue, Green alpha values variation with PSNR

Fig. 17 has shown the red, green, blue alpha values variation with the PSNR value. Red, blue and green values represented from red, blue and green bars respectively. Above figure illustrated that most of the times the alpha value of the blue component is of a higher value than other components. This observation also agreed with the Human Visual System (HVS) proof that blue component of the image catch the lesser attention of the image. Which means changes of the blue component may lesser noticeable to human eye than the green and red components.

Fig. 18 to Fig 24 has depicted the inputs and outputs of the watermarking process for the optimal alpha value combination of each attack. It shows that our watermarking approach has given reasonable robustness against most of the attacks. Damage has been done to the watermark image by the attacks is reasonably low hence we can extract the watermark image with a good quality. However, when it comes to

Butterworth attacks watermark image has destroyed considerable amount. Extracted watermark image has significant damage and the extracted image is barely recognizable. Although the watermark image is significantly distorted, the Butterworth attack has done the same kind of damage to the watermarked image as well. Hence, we can still suggest the proposed approach as an acceptable.



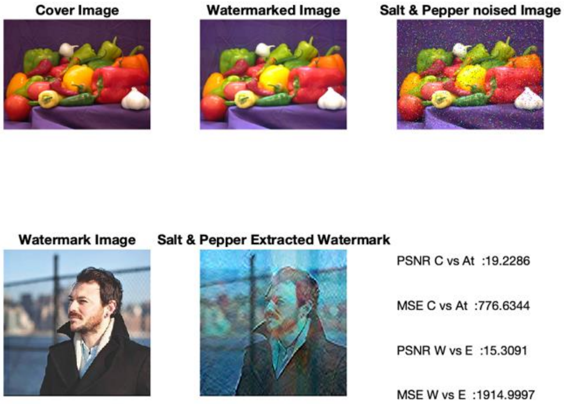
PSNR C vs At :20.7809

MSE C vs At :543.2436

PSNR W vs E :15.2651

MSE W vs E :1934.4888

Fig. 18 Resulting images using optimal (0.7,0.6,0.8) alpha values in gaussian noise attack.



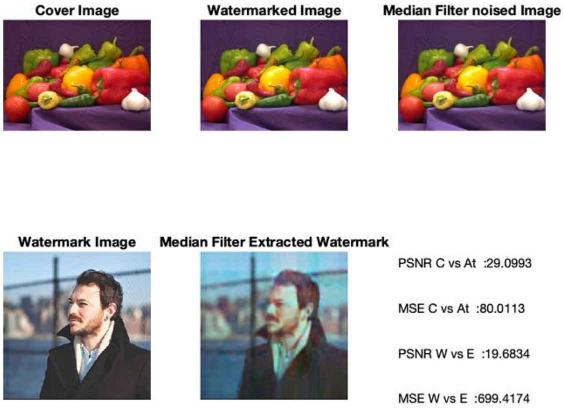
PSNR C vs At :19.2286

MSE C vs At :776.6344

PSNR W vs E :15.3091

MSE W vs E :1914.9997

Fig. 19 Resulting images using optimal (0.7,0.9,0.9) alpha values in salt & pepper attack.



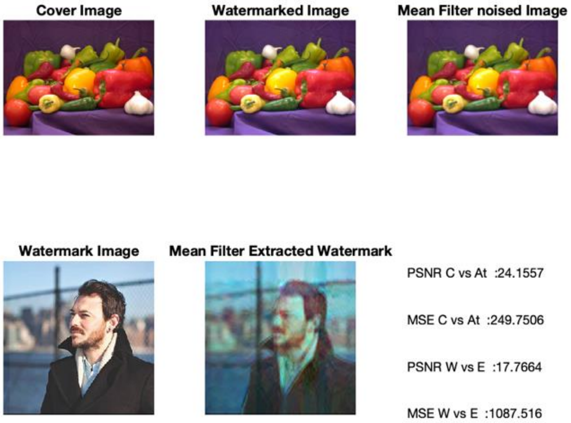
PSNR C vs At :29.0993

MSE C vs At :80.0113

PSNR W vs E :19.6834

MSE W vs E :699.4174

Fig. 20 Resulted images using optimal (0.1,0.3,0.3) alpha values in median filter attack.



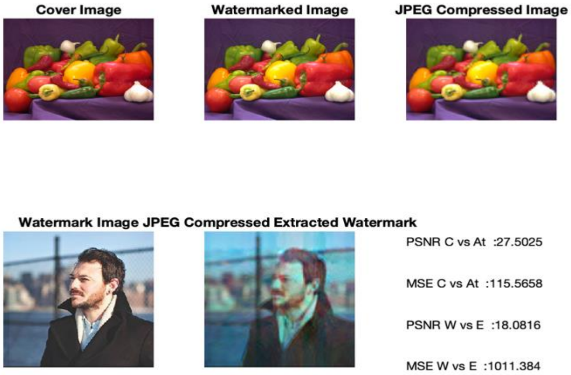
PSNR C vs At :24.1557

MSE C vs At :249.7506

PSNR W vs E :17.7664

MSE W vs E :1087.516

Fig. 21 Resulted images using optimal (0.3,0.4,0.6) alpha values in mean filter attack.



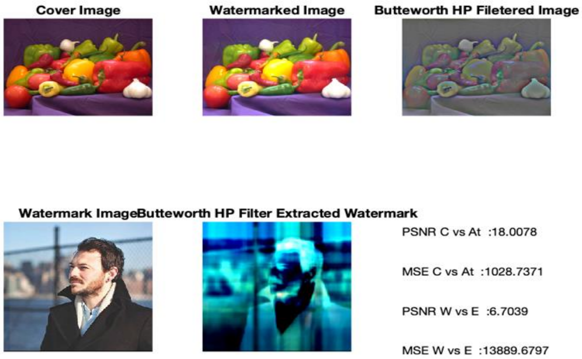
PSNR C vs At :27.5025

MSE C vs At :115.5658

PSNR W vs E :18.0816

MSE W vs E :1011.384

Fig. 22 Resulted images using optimal (0.2,0.4,0.3) alpha values in JPEG compression attack.


Cover Image
Watermarked Image
Butteworth HP Filetered Image
Watermark ImageButteworth HP Filter Extracted Watermark

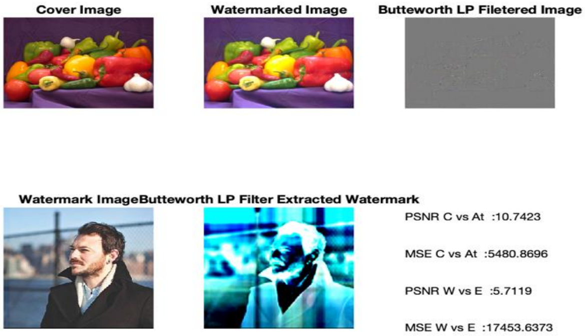
PSNR C vs At :18.0078

MSE C vs At :1028.7371

PSNR W vs E :6.7039

MSE W vs E :13889.6797

Fig. 23 Resulted images using optimal (1.0,1.0,1.0) alpha values in Butterworth high-pass filter attack.


Cover Image
Watermarked Image
Butteworth LP Filetered Image
Watermark ImageButteworth LP Filter Extracted Watermark

PSNR C vs At :10.7423

MSE C vs At :5480.8696

PSNR W vs E :5.7119

MSE W vs E :17453.6373

Fig. 24 Resulted images using optimal (1.0,1.0,1.0) alpha values in Butterworth low-pass filter attack.

Here we have come up with the specific alpha value combination for each type of attack. But we require a more generalized solution which is suitable for any kind of the attack as well as the attack-less situations.

On without attack situations image deviation between original watermark and extracted watermark did not changed according to the alpha value combinations. Deviation between cover image and watermarked image only differs according to alpha values. Hence, for the calculation of generalized value for all situation we have considered deviation between cover and watermarked image of without attacks situations and deviation between watermark and extracted watermark for attack situations. We take the PSNR value variation of each above mentioned situations and analysed them in order to find a generalized alpha value combination.

Here onward we have explained how to calculate the generalized alpha values set. We have selected following data fields to calculate the generalized alpha values set.

- PSNR between cover vs watermarked image in without any attack.
- PSNR between watermark vs extracted watermark image against Gaussian noise attack.
- PSNR between watermark vs extracted watermark image against Salt & Pepper attack.
- PSNR between watermark vs extracted watermark image against Median filter attack.
- PSNR between watermark vs extracted watermark image against Mean filter attack.
- PSNR between watermark vs extracted watermark image against JPEG compression attack.
- PSNR between watermark vs extracted watermark image against Butterworth high pass filter attack.
- PSNR between watermark vs extracted watermark image Butterworth low pass filter attack.

As the first step we calculated mean and the standard deviation of each row which had taken to the calculations. The Table V shown the calculated results.

TABLE V
CALCULATED MEAN AND STANDARD DEVIATION FOR EACH COLUMN.

Attack Type	Mean	Standard Deviation
Without Attacks	**22.707257**	**2.728362569**
Gaussian Noise Attack	13.8435026	1.340501433
Salt and Pepper Attack	**13.2439813**	**1.722956471**
Median Filter Attack	18.041697	0.75662693
Mean Filter Attack	**16.885489**	**0.528859425**
JPEG Compression Attack	17.272446	0.427591227
Butterworth High-Pass Filter Attack	**5.5107111**	**0.477158847**
Butterworth Low-Pass Filter Attack	4.883672	0.263098637

Then we standardized each value in these data set row by row. By using following formula.

$$z = \frac{x - \mu}{\sigma}$$

μ = mean value of each column, σ = standard deviation for each column, where z given the standardized value. Then we calculated new columns with standardized values. Table VI and VII depicts the part of standardized data table.

TABLE VI
STANDARDIZED VALUES EACH COLUMN.

R Alpha	G Alpha	B Alpha	Without Attacks	Gaussian Noise Attack	Salt and Pepper Attack	Median Filter Attack
0.1	0.1	0.1	5.38665321	-3.4004459	-3.096527	1.38285192
0.1	**0.1**	**0.2**	**4.16320878**	**-2.6919051**	**-2.6275076**	**1.7886001**
0.1	0.1	0.3	3.15344562	-2.6076083	-2.4377176	1.82428479

TABLE VII
STANDARDIZED VALUES EACH COLUMN.

R Alpha	G Alpha	B Alpha	JPEG Compression Attack	Mean Filter Attack	Butterworth High-Pass Filter Attack
0.1	0.1	0.1	-0.9341773	-3.3061508	-2.1047731
0.1	**0.1**	**0.2**	**-0.6582128**	**-2.2170145**	**-2.0574094**
0.1	0.1	0.3	-0.6137778	-1.9882202	-1.9769331

Then we considered above standardized values set and generated mean value and standard deviation for each column. Then we divided mean by the standard deviation for each row and took the maximum value as the optimal alpha combination. Following of the part of generated data for the above calculation.

TABLE VIII
MEAN / STANDARD DEVIATION FOR OPTIMAL ALPHA VALUE COMBINATION.

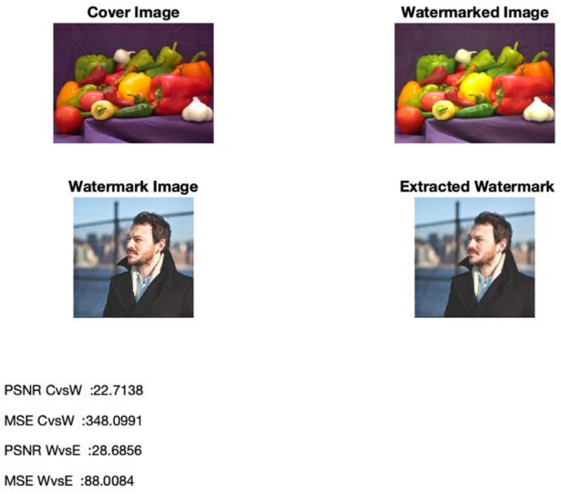
R Alpha	G Alpha	B Alpha	Mean	Sd	Mean / Sd
0.6	**0.7**	**1**	**14.2012875**	**0.91073721**	**15.593178**
0.6	0.8	0.1	13.4593375	0.5446826	24.7104232
0.6	**0.8**	**0.2**	**14.0753**	**0.17407591**	**80.8572524**
0.6	0.8	0.3	14.25675	0.3157713	45.1489728

As per the above Table VIII following is the generalized alpha value combination which is suited for every kind of situations.

TABLE VIII
OPTIMAL ALPHA VALUES COMBINATION FOR EVERY SITUATIONS.

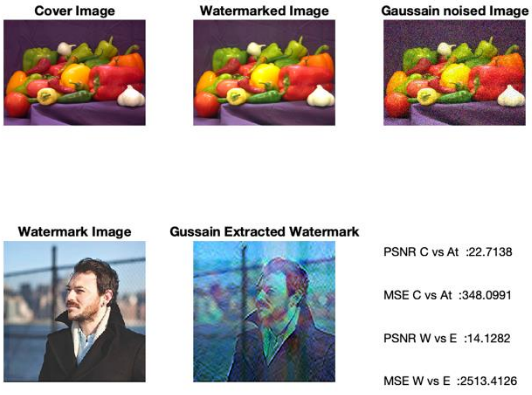
Red Alpha	**0.6**
Green Alpha	0.8
Blue Alpha	**0.2**

Fig. 25 to 32 illustrates the generated image results by applying watermarking and watermark extraction algorithm for above optimal alpha value combination.



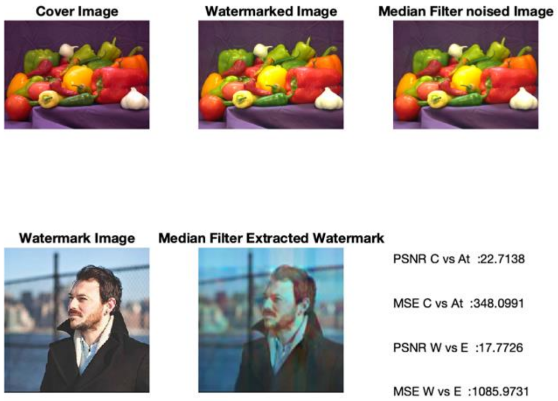
PSNR CvsW :22.7138
MSE CvsW :348.0991
PSNR WvsE :28.6856
MSE WvsE :88.0084

Fig. 25 Resulted images using generalized alpha values for without any attack executing.



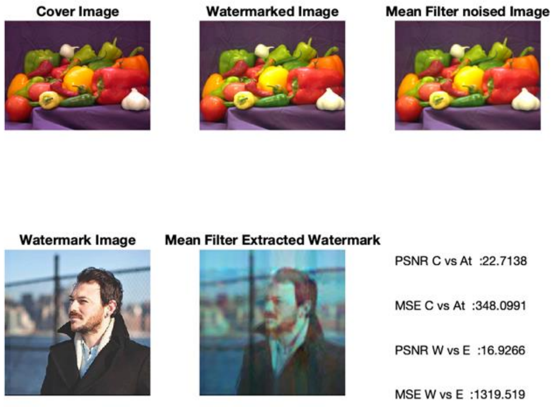
PSNR C vs At :22.7138
MSE C vs At :348.0991
PSNR W vs E :14.1282
MSE W vs E :2513.4126

Fig. 26 Resulted images using generalized alpha values for Gaussian noise attack.



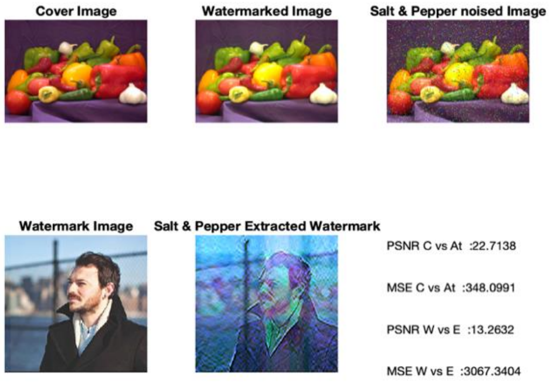
PSNR C vs At :22.7138
MSE C vs At :348.0991
PSNR W vs E :17.7726
MSE W vs E :1085.9731

Fig. 27 Resulted images using generalized alpha values for Median filter attack.



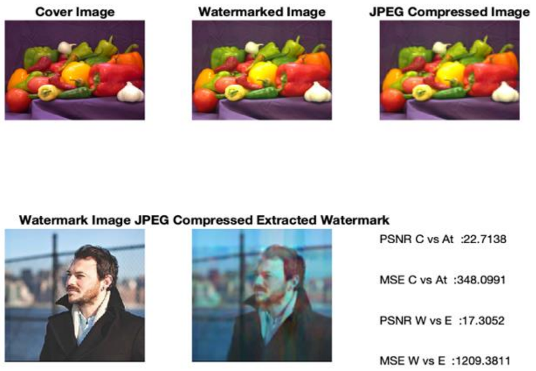
PSNR C vs At :22.7138
MSE C vs At :348.0991
PSNR W vs E :16.9266
MSE W vs E :1319.519

Fig. 28 Resulted images using generalized alpha values for Mean filter attack.



PSNR C vs At :22.7138
MSE C vs At :348.0991
PSNR W vs E :13.2632
MSE W vs E :3067.3404

Fig. 29 Resulted images using generalized alpha values for Salt & Pepper attack.



PSNR C vs At :22.7138
MSE C vs At :348.0991
PSNR W vs E :17.3052
MSE W vs E :1209.3811

Fig. 30 Resulted images using generalized alpha values for JPEG Compression attack.

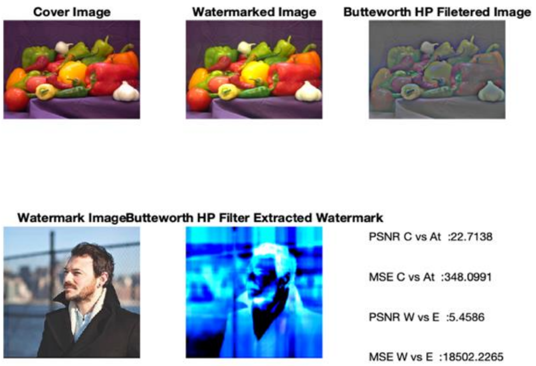


Fig. 31 Resulted images using generalized alpha values for Butterworth High-pass Filter attack.

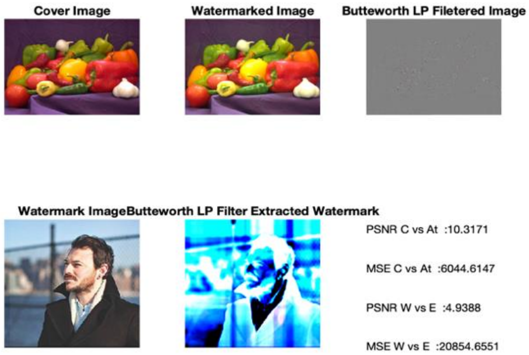


Fig. 32 Resulted images using generalized alpha values for Butterworth Low-pass Filter attack.

Table IX has summarized the PRNR and MSE measures for each attack type for generalized alpha values combination.

TABLE IX
PSNR, MSE VALUES FOR EACH ATTACK TYPE USING GENERALIZED ALPHA VALUES COMBINATION.

Attack Type	Cover vs Watermarked		Watermarked vs Extracted	
	PSNR	MSE	PSNR	MSE
Without Attacks	22.714	348.1	28.686	88.008
Gaussian Noise Attack	18.487	921.32	14.141	2506.2
Salt and Pepper Attack	16.538	1443.1	13.345	3010.1
Median Filter Attack	22.653	352.97	17.773	1086
Mean Filter Attack	22.714	348.1	16.927	1319.5
JPEG Compression Attack	22.077	403.11	17.305	1209.4
Butterworth High-Pass Filter Attack	13.421	2958.1	5.4586	18502
Butterworth Low-Pass Filter Attack	10.317	6044.6	4.9388	20855

To get an idea about how much results have been deviated from their individual optimal values when using generalized alpha values combination, data for both situations needs to be analysed. The summery of data for both situations are presented in Table X.

TABLE X
PSNR, MSE VALUES COMPARISON FOR BOTH GENERALIZED ALPHA VALUES COMBINATION AND INDIVIDUAL OPTIMAL ALPHA VALUES COMBINATION.

Attack Type	Generalized alpha values combination				Individual optimal alpha values combination			
	Cover vs Watermarked		Watermarked vs Extracted		Cover vs Watermarked		Watermarked vs Extracted	
	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Without Attacks	22.714	348.1	28.686	88.008	37.404	11.823	28.686	88.008
Gaussian Noise Attack	18.487	921.32	14.141	2506.2	17.648	1117.6	15.237	1947.2
Salt and Pepper Attack	16.538	1443.1	13.345	3010.1	15.499	1833	15.382	1883.1
Median Filter Attack	22.653	352.97	17.773	1086	28.78	86.118	19.683	699.42
Mean Filter Attack	22.714	348.1	16.927	1319.5	24.156	249.75	17.766	1087.5
JPEG Compression Attack	22.077	403.11	17.305	1209.4	25.735	173.61	18.082	1011.4
Butterworth High-Pass Filter Attack	13.421	2958.1	5.4586	18502	13.323	3025.6	6.7039	13890
Butterworth Low-Pass Filter Attack	10.317	6044.6	4.9388	20855	10.742	5480.9	5.7119	17454

Fig. 33 illustrates the PSNR variation of both occasions in cover image and the watermarked image. The blue bars of the graph depicted the PSNR values for generalized alpha values combination and orange bars of the graph showed the PSNR values for individual optimal alpha values combination for each attack.

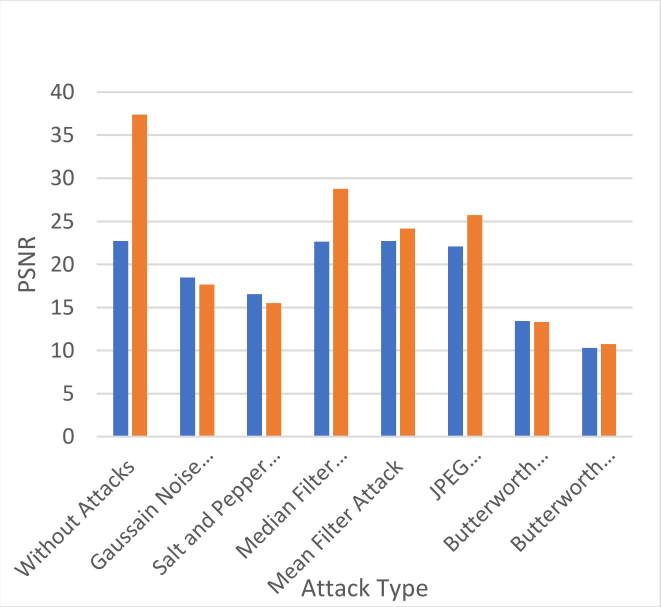


Fig. 33 Difference between PSNR values of cover vs watermarked images in both occasions.

According to Fig. 33 it could be concluded that for the higher PSNR values the difference is noticeable. But for the lesser PSNR values very small difference is there. We have also drafted the PSNR values variation of original watermark image and the extracted watermark image. Following Figure 34 showed that variation. In that figure also blue bars depicted the PSNR values of the generalized alpha values set and orange bars showed the PSNR of the individual optimal alpha values combination. PSNR values has changed from very small amounts in every attack type. We could see that when we take the watermark image and the extracted watermark difference into the consideration generalized alpha values set make much more sense, because difference of the PSNR are very low as we can remove that difference from our consideration.

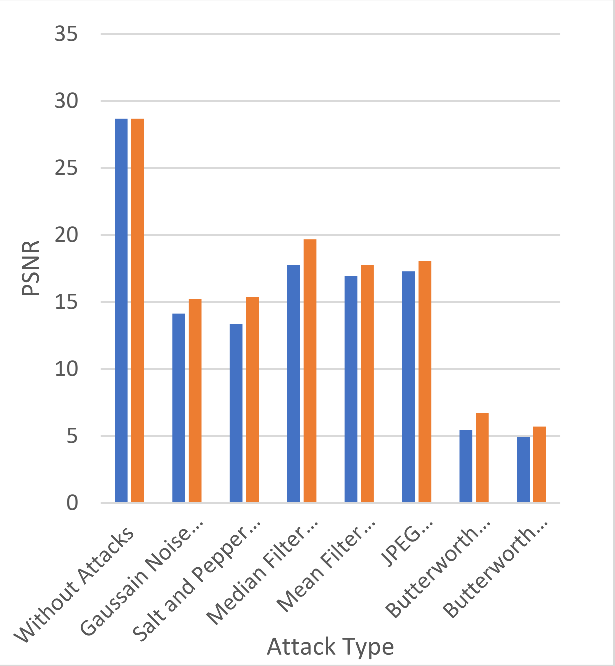


Fig. 34 Difference between PSNR values of watermark and extracted watermark images in both occasions.

Table XI has summarized the difference of PSNR values of cover vs watermarked and watermark versus extracted images in both occasions. Here we could see that cover versus watermarked image deviation sometimes higher in the individual optimal value scenario than the generalized alpha values combination. The reason for that is we have only considered deviation between original image and the extracted watermark image in the attack occurring instances.

TABLE XI
PSNR VALUES DIFFERENCE IN GENERALIZED ALPHA VALUES AND INDIVIDUAL OPTIMAL ALPHA VALUES FOR EACH ATTACK TYPE.

Attack Type	Difference of PSNR Values	
	cover vs watermarked	watermark vs extracted
Without Attacks	14.69	0
Gaussian Noise Attack	-0.839	1.096
Salt and Pepper Attack	-1.039	2.037
Median Filter Attack	6.127	1.91
Mean Filter Attack	1.442	0.839
JPEG Compression Attack	3.658	0.777
Butterworth High-Pass Filter Attack	-0.098	1.2453
Butterworth Low-Pass Filter Attack	0.425	0.7731

Fig. 35 represents the PSNR value difference between generalized alpha values combination and the individual optimal alpha values combinations. The blue bars showed the deviation between cover image and watermarked image, orange bars showed the deviation between original watermark image and the extracted watermark image.

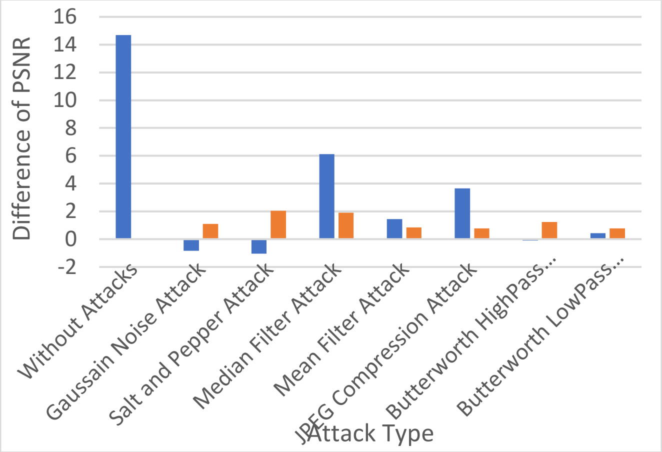


Fig. 35 Difference of PSNR values in generalized alpha values combination and individual optimal values combination.

In this figure we could see that the difference between cover versus watermarked image is significant. However, the difference between watermark image and the extracted watermark image is not much significance. Hence, it can be concluded that generalized alpha value combination also gives reasonably good results.

D. Capacity

Capacity is also one of the main properties watermarking algorithm. A watermarking algorithm should provide a facility to embed a reasonable sized image as a watermark. Embedding a watermark image which is larger than the cover image may not possible. But at least algorithm should provide an opportunity to embed a 50% - 75% sized image as the cover image. In this section we have described an experiment to test the capacity of the suggested approach and discussion and evaluation about the results.

1) Experiment

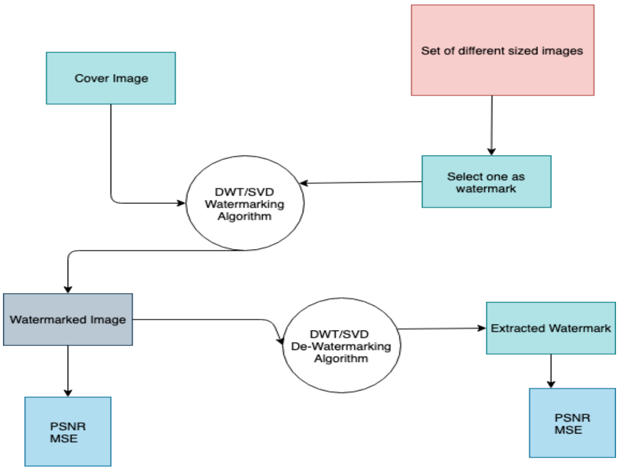

Set of different sized images
Cover Image
Select one as watermark
DWT/SVD Watermarking Algorithm
Watermarked Image
DWT/SVD De-Watermarking Algorithm
Extracted Watermark
PSNR MSE
PSNR MSE

Fig. 36 Experiment design for capacity evaluation.

Fig. 36 has shown the design of the suggested experiment. Here we have used peppers.png (96 × 128) of the MATLAB toolbox as a cover image. Then we have selected different sized MATLAB toolbox images with a 0.1, 0.1,0.1 alpha value combination which is the optimal alpha value combination for the without attack situations. We have selected one image per time and execute the watermarking and watermark extraction process separately. PSNR values of the watermarked image and the extracted image will be recorded.

2) Results Analysis and Evaluation

In this section we have presented and evaluate the data that we have captured in the above experiment. Table XII has depicted the captured data of the experiment.

TABLE XII
CAPTURED DATA FROM CAPACITY EXPERIMENT.

image name	Cover Image		Watermark Image		Ratio (%)	Cover vs Watermarked		Watermark vs Extracted	
	Wid.	Hei.	Wid.	hei.		PSNR	MSE	PSNR	MSE
onion.png	512	384	198	135	13.59	39.75	6.88	37.38	11.884
hestain.png	512	384	303	227	34.98	35.26	19.34	30.81	53.931
football.jpg	512	384	320	256	41.66	41.61	4.48	29.42	74.161
Gantrycr-ane.png	512	384	400	264	53.71	38.92	8.32	22.56	359.98
Westconco-rdaerial.png	512	384	369	394	73.94	37.16	12.47	25.38	188.28
greens.jpg	512	384	500	300	76.29	42.39	3.74	20.27	609.96
pillsetc.png	512	384	512	384	100	40.88	5.30	25.19	196.44
peppers.png	512	384	512	384	100	40.64	5.60	31.00	51.569
tape.png	512	384	512	384	100	42.23	3.88	29.85	67.294
fabric.png	512	384	640	480	156.25	40.49	5.79	21.68	441.13
pears.png	512	384	732	486	180.94	37.80	10.77	31.73	43.578
tissue.png	512	384	800	506	205.89	34.89	21.08	17.00	1296.7
saturn.png	512	384	903	600	275.57	42.30	3.82	26.25	154.1
saturn.png	512	384	1200	1500	915.53	43.52	2.88	34.61	22.466
Concorda-erial.png	512	384	3060	2036	3168.8	38.92	8.32	21.19	493.38

We have drafted PSNR change against capacity of the watermark image in Fig 37. The blue colour graph showed the deviation between cover image and watermarked image and orange one showed the deviation between watermark image and the extracted watermark.

Here we couldn't see much deviation according to the ratio. But we can assure that 0% - 200% ration range have achieved the better results. Suggested approach has been able to achieve the capacity more than 100%. That is a significance of that approach where there wasn't any watermarking approach achieve this capacity level.

V. CONCLUSION

A. *Conclusion*

In this research we introduced a watermarking algorithm for colour images. Our aim was to build a watermarking algorithm which can achieved the main conflicting properties of good watermarking algorithm. Those are imperceptibility, capacity and robustness. Simply the imperceptibility means the embedding process should not deviate the original image in a noticeable amount.

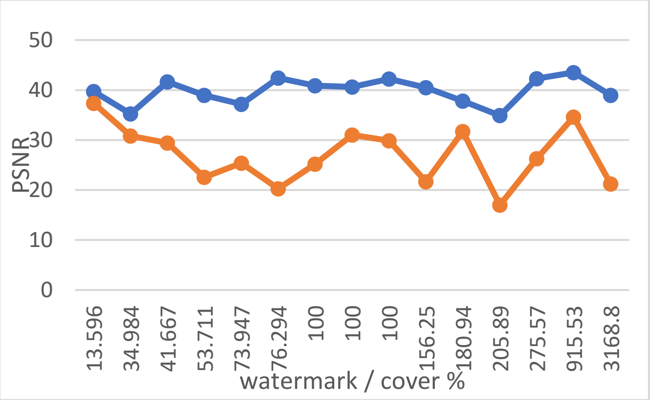


Fig. 37 PSNR of cover vs watermarked and watermark vs extracted variation with the capacity of watermark image.

Both the deviation between cover and watermarked image and the deviation between watermark image and the extracted watermark image can be put under imperceptibility. This is the main visible requirement of this algorithm. Other requirements such as capacity and robustness are not directly visible to the human eye. There were the solutions which achieved one or two among these requirements, but none of those were able to achieve all three requirements together in a single solution. Hence, aim of this research was to come up with a solution which can achieve all three requirements together.

According to the evaluation which we have done, for the alpha value set 0.6, 0.8, 0.2 for red, blue, green respectively gives the higher robustness for all kind of attacks and attack-less situation. This value set was able to achieve all the three basic watermarking requirements together. We have achieved deviation between cover and watermarked image 22.714 to 10.317 PSNR value for every kind of attack. Table IX showed the full details of PSNR and MSE values of generalized alpha value combination. Which means our algorithms was able to achieve both imperceptibility and robustness requirements together up to considerable amount. According to the Fig. 37 we can embed the watermark image up to 200% size which compared to the size of the cover image. This is a significance achievement because we have passed the 100% margin as well. Which means we can use the double sized watermark image with respect to the size of the cover image.

We can conclude that suggested algorithm was able to achieve all the three basic requirements together. Which is a significance achievement. That make this algorithm more efficient than all other existing algorithms. When we use optimized alpha value combination for each type of attack as shown as the Table IV. We can use this algorithm against specific type of attacks as well. Those optimized alpha value combination gives better results than the generalized alpha value combination.

B. *Limitations and Future Work*

However, there are few limitations of this approach. The configurations of the cover image and the watermark image should be alike when considering the watermarking process. Configurations such as byte rate should be equal of both images. In our experiments, most of the times we used 8-bit image. There is a possibility to improve this algorithm for working as images with a different byte rate as well. For that, we need to convert the byte rate of the watermark image programmatically to match with the byte rate of the cover image first. Then apply the existing watermarking algorithm and the watermark extraction algorithm without any issues. After the extraction process we have to convert extracted watermark to its original byte rate. Those kinds of improvement would be possible to implement for this algorithm.

This approach is a non-blind one. It is better if we can improve this as a blind watermarking technique as well. Our approach required one of the SVD decomposed matrices of cover image to execute the watermark extraction process. It's better if we can execute the watermark extraction process without taking any components of the cover image as well. To achieve that we need some considerable modifications for the existing equations of the existing algorithm. We need to find a way to recognize embedded data without help of the original cover image. For that we have to mark the embedded data using some sort of mechanism which can be separately recognize the data of the original cover image.

Robustness against attacks can be further improved as well. There is nothing wrong if we can improve the robustness of the watermark as much as possible. We have witnessed that attacks like Butterworth filtering has doing some considerable damage to the watermarked image as well as the extracted watermark. We can further improve the robustness of the watermark by applying few different solutions. Currently, we apply the DWT only once. We can increase this and may apply the DWT to the cover image few times. We could expect some reasonable robustness increment from that. Also, we can modify the formulas of current algorithm and make them more complex. In this approach we have tried to keep formulas simple, because we want to clearly observe the effect of other factors to the algorithm such as alpha values.

REFERENCES

[1] E. Lin and E. J. Delp, "A Review of Fragile Image Watermarks," 1999, pp. 25–29.

[2] Ying Yang, Xingming Sun, Hengfu Yang, Chang-Tsun Li, and Rong Xiao, "A Contrast-Sensitive Reversible Visible Image

[3] Yongjian Hu, Byeungwoo Jeon, "Reversible Visible Watermarking and Lossless Recovery of Original Images," IEEE Trans. CIRCUITS Syst. VIDEO Technol., vol. 16, no. 11, Nov. 2006.

[4] D. Pradhan, "Implementation of Invisible Digital Watermarking Technique for Copyright Protection using DWT-SVD and DCT," IJAERS Res. J., vol. 4, pp. 063–069, Jul. 2017.

[5] A. Treméau and D. Muselet, "Recent Trends in Color Image Watermarking," J. Imaging Sci. Technol., vol. 53, Jan. 2009.

[6] C. Song, S. Sudirman, and M. Merabti, "Recent Advances and Classification of Watermarking Techniques in Digital Images," p. 6.

[7] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent Robust Image Watermarking," 1996, pp. 211–214.

[8] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[9] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "A secure, robust watermark for multimedia," in Information Hiding, 1996, pp. 185–206.

[10] D. J. Fleet and D. J. Heeger, "Embedding invisible information in color images," 1997, vol. 1, pp. 532–535.

[11] G. M. Johnson and M. D. Fairchild, "A top down description of S-CIELAB and CIEDE2000," Color Res. Appl., vol. 28, no. 6, pp. 425–435, Dec. 2003.

[12] N. V. Dharwadkar and B. B. Amberker, "AN EFFICIENT AND SECURED NON BLIND WATERMARKING SCHEME FOR COLOR IMAGES USING DWT AND ARNOLD TRANSFORM," vol. 9, no. 2, p. 9, 2010.

[13] V. Santhi and A. Thangavelu, "DC Coefficients Based Watermarking Techniquefor color Images Using Singular ValueDecomposition," Int. J. Comput. Electr. Eng., pp. 8–16, 2011.

[14] H. Gao, L. Jia, and M. Liu, "A Digital Watermarking Algorithm for Color Image Based on DWT," vol. 11, no. 6, p. 8, 2013.

[15] C.-H. Chou and T.-L. Wu, "Embedding Color Watermarks in Color Images," EURASIP J. Appl. Signal Process., p. 9, 2003.

[16] P. Bas, N. Le Bihan, and J.-M. Chassery, "Color image watermarking using quaternion Fourier transform," 2003, vol. 3, pp. III-521–4.

[17] C.-S. Chan, "Color image hiding scheme using image differencing," Opt. Eng., vol. 44, no. 1, p. 017003, Jan. 2005.

[18] S. P. Mohanty, P. Guturu, E. Kougianos, and N. Pati, "A Novel Invisible Color Image Watermarking Scheme Using Image Adaptive Watermark Creation and Robust Insertion-Extraction," in Eighth IEEE International Symposium on Multimedia (ISM'06), 2006, pp. 153–160.

[19] P. A. Kaarna and V. Botchko, "A Technique for Digital Color Image Watermarking Using ICA," p. 53, 2006.

[20] R. Agarwal and K. Venugopalan, Digital Watermarking of Color Images in the Singular Domain. 2011.

[21] "Digital Color Image Watermarking using DWT-DCT Coefficients in RGB Planes | Global Journal of Computer Science and Technology," 2013. [Online]. Available: https://computerresearch.org/index.php/computer/article/view/175. [Accessed: 13-Aug-2018].

[22] D. Vaishnavi and T. S. Subashini, "Robust and Invisible Image Watermarking in RGB Color Space Using SVD," Procedia Comput. Sci., vol. 46, pp. 1770–1777, 2015.

[23] J. guru and H. damecha, "A Review of Watermarking Algorithms for Digital Image," vol. 2, no. 9, p. 8, 2007.

[24] G. Chawla, R. Saini, and R. Yadav, "Classification of Watermarking Based upon Various Parameters," Int. J. Comput. Appl., p. 4, 2012.

[25] S. Madhesiya and S. Ahmed, "Advanced Technique of Digital Watermarking based on SVD-DWT-DCT and Arnold Transform," vol. 2, no. 5, p. 6.

[26] Kamal Ali Ahm, "Digital watermarking of still images," 2013.

[27] B. P. Mishra and D. H. N. Pratihari, "DCT Based Grey Scale Still Image Watermarking Using 1-D Walsh Code and Biometric Protection," vol. 4, no. 2, p. 5, 2015.