A Coloured Image Watermarking Based on Genetic K-Means Clustering Methodology

Zainab Falah Hassan, Farah Al-Shareefi, and Hadeel Qasem Gheni*

Department of Computer Science, Babylon University, Babylon, Iraq
*Correspondence: hajer.s.abbas.uoesraa@gmail.com (H.Q.G.)

***Abstract*—There are two techniques long-established in image watermarking area, namely the k-means and genetic algorithms. The first one is commonly used to allocate an image's pixels into distinct clusters. However, the allocation of these pixels is not optimal in all cases. The second technique is usually employed to produce an optimal watermarking solution. In this paper, a hybrid methodology is presented for coloured image watermarking that integrates both genetic algorithm and k-means clustering activity to attain the optimized cluster centroids. These centroids are utilized to optimally distribute the pixels of the cover and watermark images into suitable clusters. This will help decrease the perceptible changes in the watermarked image with the naked eye. For concealment, the Least Significant Bits method is adopted. Typically, the pixels of every watermark cluster are concealed in its closest cover's cluster; wherein every two successive pixels hide the bits of a single cover image's pixel. The experimental results demonstrate that the proposed methodology satisfies a sufficient imperceptibility that yields and boosts resistance against common attacks.**

***Keywords*—genetic algorithm, LSB method, K-means clustering, watermarking techniques, attack types, performance measures**

I. Introduction

Due to the tremendous increase in access to digital media content, protection issues related to the ownership and copyright of published media have become an urgent demand. To this extent, *digital watermarking* technology appears as an effective technique to tackle these issues [1]. In fact, digital watermarking is a technique for hiding data in different content carriers, such as audio [2], video [3], code [4], and image [5]. As for the context of this paper, coloured image watermarking is considered.

Digital watermarking methods are classified according to the used domain into *frequency domain methods* that embed the hidden data in the frequency representation of the cover image, and *spatial domain methods* that hide data into image pixels [6]. This paper focuses on the spatial method.

In the image watermarking field, there are two techniques commonly used: *k-means clustering* and *genetic algorithm*. The first one is an unsupervised partitioning technique that is performed in an iterative manner [7]. It was originally designed to assign numerical data into clusters with the closest centroids. Nevertheless, the greedy assignment of data points would not be optimal in all cases. The non-optimal clustering process leads to an increase in the apparent changes in the cover image.

On the other hand, the second technique of the genetic algorithm is an evolutionary search algorithm directed by the rules of natural genetics to find the optimal solution to an optimization problem [8]. As a result, the work of this paper is motivated to combine the optimal performance of the genetic algorithm with the clustering process of the k-means technique, in order to identify the optimum clusters for embedding the watermark information using colour images. To perform the embedding process, the Least Significant Bit (LSB) method is applied [9]. Essentially, the pixels of every watermark image's cluster are embedded in its closest cover image's cluster. The purpose of this concept is to increase both the security and imperceptibility levels of the watermarking outputs. In addition, the last four bits of two consecutive pixels in a cluster belonging to the cover image are changed with the bits of a pixel in a cluster related to the watermark image. Using the closest clusters helps to create a balance between the change in pixel value, which can be noticed by the human eye.

The main contributions of this paper are:

1) The proposal of a simple colour image watermarking methodology that hybridizes genetic algorithm with k-means clustering to raise both watermarking requirements: robustness and imperceptibility.
2) The replacement of the iterative style of the k-means technique by that of the genetic algorithm to accelerate the convergence to the best watermarking solution.

The rest of this paper is structured as follows. Section II highlights the most relevant related work. In Section III, the utilized materials and methods are briefly explained. Section IV illustrates the proposed methodology in detail. Section V discusses the obtained results. Finally, Section VI recapitulates this paper.

II. Related Works

This section states the related works that utilize the genetic algorithm, clustering techniques, and LSB method

Manuscript received June 9, 2022; revised August 10, 2022, accepted October 19, 2022; published March 17, 2023.

for image watermarking. An enhanced LSB schema is developed for image watermarking [10]. The main idea of the proposed schema is to arbitrarily embed the watermark bits in the cover image coordinates by utilizing a random mapping function. The developed schema helps to improve the robustness of the watermarking process.

A colour image watermarking technique is presented in [11]. In this technique, the cover image is partitioned into k clusters relying on the cluster indices. These indices are further partitioned into sub-clusters depending on the pixel values in the red, green, and blue components of the cover image. The sub-clusters are employed to embed the pixels of the watermark image. The obtained results have shown that the developed technique is capable of resisting rotation attacks. However, the results have not shown the efficiency of this technique against other attack types, such as noise attacks.

A steganography algorithm based on the genetic algorithm is proposed in [12] for hiding text or image data in a colour image. In essence, the genetic algorithm directs the steganography process to the best locations for hiding the data within the cover image. Though the proposed algorithm helps to gain a small distortion rate for the steganography process, this rate inversely correlates with the cover images' details, i.e., it increases when the used cover image is a low-detail one. In addition, the execution time here is directly proportional to the size of the concealed data.

Different from the focus suggested in the present work, the authors in [13] designed a verification method for digital images by using public-key cryptography and the LSB method. Through the designed method, the encrypted watermark image is concealed in the least significant bits of the cover image in order to avoid tampering with the watermark information. In addition, the authors in [14] proposed a tampered watermarked image detection method based on the k-means clustering technique. They form clusters depending on 0's and 1's in the least significant bits of the original image. The watermark bits are embedded in the formed clusters. The clustering process is implemented for both the original watermarked image and the suspected watermarked one to detect the tampering action.

III. Materials and Methods

In this section, the main materials and methods utilized in this paper are briefly described.

A. Least Significant Bit Method

In digital watermarking, the most popular technique to conceal data in an image is the *Least Significant Bit* (*LSB*) method [9]. It has gained popularity due to its simplicity and undetectability with the naked eye. In principle, LSB embeds the binary secret data into the last bit; it is also called the least significant bit of each byte or bytes per pixel. It should be noted that every pixel in grey images is represented by a byte, meanwhile, its representation requires three bytes in the coloured ones. The least significant bit is used to dramatically reduce the variations in the colours of an altered image. In other words, the altered image becomes indistinguishable from the original one when hiding the data within a particular bit.

B. Genetic Algorithm

A Genetic Algorithm (GA) is a search algorithm designed to solve optimization problems by imitating the main concepts of natural biological inheritance [15, 16]. Essentially, this algorithm reflects the Darwinian evolutionary theory as it selects the fittest individuals (the elected solutions) from a population to produce the next generations (the most optimized solutions). More precisely, GA commences by generating an arbitrary population of individuals known as chromosomes. Next, these individuals are graded in conformity with their fitness values (objective function values). After that, the next generation is obtained by applying the following operators: selection, crossover, and mutation. Selection refers to selecting the fittest individuals in order to allow them to give their genes or the binary values that constitute a chromosome, to the next population. Crossover is a reproduction operation that combines the gene values of two selected individuals to create a new generation. Mutation flips some gene values in the obtained generation to maintain its diversity. The three operators are repeatedly applied to every gained generation until convergence is attained.

C. K-Means Clustering Technique

One of the most straightforward and well-known clustering techniques is k-means clustering [17, 18]. It aims at aggregating resemblant data points into distinct groups in order to identify their specific structures. K-means have been successfully applied to signal processing and digital watermarking.

Theoretically, two essential terminologies are used with this technique: *cluster* and *centroid*. A group of data points that are collected together according to a certain shared homogeneous feature is called a *cluster*, whereas the center of that cluster is known as a *centroid*. In mathematical terms, a centroid denotes an arithmetic mean (an average) of the data points within a cluster. The number of clusters is manually determined by a user and assigned to the variable k. Based on the aforementioned aspect, k-means tries to allocate every data point to the cluster whose centroid gives the minimum distance (Manhattan Distance, Euclidean Distance, etc.) between it and that data point. The k-means algorithm can be iteratively conducted through the following steps:

1) The cluster centroids are indicated by the arbitrary selection of k points as the initial centroids.
2) The following steps are iterated until the clusters converge; i.e., there is no change to the allocation of data points to clusters:
   a) A similarity or distance is computed between every data point and all centroids.
   b) Every data point is allocated to the closest cluster according to the minimum computed distance.
   c) The new centroids are evaluated for the obtained clusters by computing the mean of every cluster.

IV. Proposed Methodology

The proposed methodology consists of two phases: the *concealment* phase and the *extraction* phase. The concealment phase is dedicated to hiding the digital watermark into the cover image through the implementation of five stages. Meanwhile, the extraction phase is directly conducted via one stage to retrieve the original watermark. The concealment phase of the proposed methodology is illustrated in Fig. 1 below in detail.

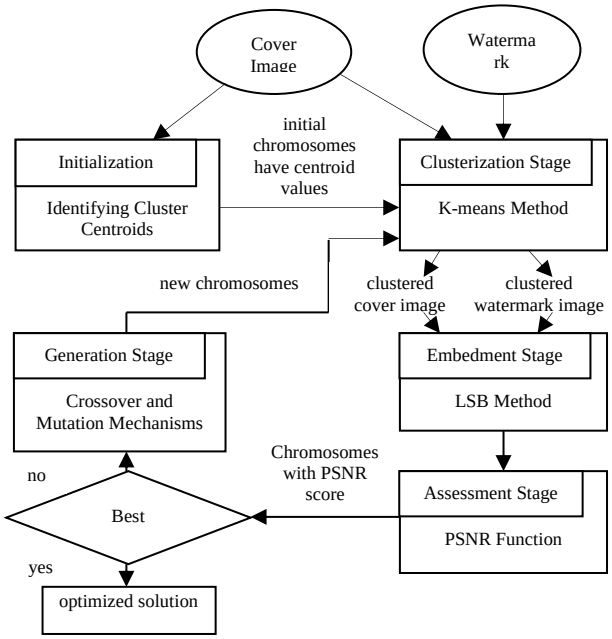


Figure 1. The proposed concealment phase.

A. Concealment Phase

The five stages of the concealment phase are as follows:

1) *Initialization stage*: This stage involves the initial process for populating the elementary chromosomes by combining the first steps of both genetic algorithm and k-means method.
2) *Clusterization stage*: It is a partitioning task of both the cover and watermark images via the k-means method, depending on the generated chromosome.
3) *Embedment stage*: It is a process at which the watermark information is enclosed in the cover image, thereby applying the core idea of the LSB method.
4) *Assessment stage*: It is an evaluation step for determining the best produced chromosomes by PSNR measure.
5) *Generation stage*: It is the production process of a new population of chromosomes through the application of the remaining steps of the genetic algorithm.

Fig. 1 shows an overview of these stages which can be clarified in more detailed subsections.

1) Initialization stage

At this stage, it is proposed to populate N of the initial chromosomes with the aid of k-means clustering. In other words, the initial steps of both the genetic algorithm (generating an initial population of chromosomes) and the k-means method (indicating the cluster centroids) are simultaneously applied to the cover image in order to assign the arbitrarily selected cluster centroids to each chromosome. These are an advantageous step as they increase the security level of the watermarking process, and they enhance the overall performance of the proposed methodology. The steps for carrying out this stage are described as follows:

1) The clusters' number is identified so as to be assigned to k.
2) Let $C_{h\times w}$ be a cover image with h and w dimensions.
3) The $C_{h\times w}$ is split into three component colours: red, green, and blue—each with their equivalent matrix $RC_{h\times w}$, $GC_{h\times w}$, $BC_{h\times w}$.
4) From each colour matrix, cluster centroids $c_{i,j}$, are randomly selected, whereby $i = 1, 2, 3$ and $j = 1, \ldots, k$, in order to compose the initial chromosome.

A chromosome is encoded as a $3\times k$ matrix, so that 3 is the number of the basic image colours: red, green, blue, and k is the assumed number of clusters. Fig. 2 illustrates the representation of a chromosome.

$c_{1,1}$	$c_{1,2}$	$\cdots$	$c_{1,k}$
$c_{2,1}$	$c_{2,2}$	$\cdots$	$c_{2,k}$
$c_{3,1}$	$c_{3,2}$	$\cdots$	$c_{3,k}$

where $c_{i,j}$ *is a cluster centroid at* (i,j) *coordinate,* $i = 1,2,3,\ \ j = 1,\ldots,k$

Figure 2. Structure of a chromosome.

It is worth noting that every row of a chromosome-matrix is filled with cluster centroids that belong to a specific cover image colour. The first row is allocated to the centroids of clusters related to the red colour, while the second and the third rows belong to the green and blue colours, respectively.

2) Clusterization stage

This stage employs the k-means method to partition both the cover image and the watermark image into k clusters. Typically, the obtained selected centroids in the previous stage are utilized to organize the pixels of these images into clusters. The organization process is accomplished by using the *Manhattan* distance metric [19], as it gives effective results in measuring similarity for a high-dimensional data set [20]. This metric computes the distance between two vectors: u and v of length d as follows:

$$M_D(u,v) = \sum_{i=1}^{d} |u_i - v_i| \qquad (1)$$

Accordingly, the clustering process can be conducted through the following steps:

1) The cover image is put in, and each pixel $p_{a,b}$, $\forall a = 1,\ldots,h, \forall b = 1,\ldots,w$ is assigned in each colour matrix that belongs to the input image: $RC_{h\times w}$, $GC_{h\times w}$, $BC_{h\times w}$ to cluster cl_j, so that

$$|p_{a,b} - c_{i,j}| < |p_{a,b} - c_{i,l}|, \forall l = 1, \dots, k, l \neq j, \text{ and}$$

$$i = 1 \Leftrightarrow p_{a,b} \in RC_{h \times w}$$

$$i = 2 \Leftrightarrow p_{a,b} \in GC_{h \times w}$$

$$i = 3 \Leftrightarrow p_{a,b} \in BC_{h \times w}$$

2) Let $W_{m \times n}$ be a watermark image with *m* and *n* dimensions.
3) The $W_{m \times n}$ is split into three component colours: red, green, and blue—each with their equivalent matrix $RW_{m \times n}$, $GW_{m \times n}$, $BW_{m \times n}$.
4) The first step is reapplied, but the colour matrices are swapped for the cover image with those for the watermark image. In addition, the watermark image will have cluster wl_J instead of cluster cl_j, where $J = 1, \dots, k$.

It should be noted that the clustering steps above do not involve an iterative style for producing the converged centroids, as the genetic algorithm help provide the best centroids. In addition, the clustering stage is conducted to help reduce the noise in the cover image, and this will later benefit the watermarking process in general. The reduction of noise originates from utilizing the closest clusters for concealing secret information, as these clusters lead to diminishing the changes in the cover image and subsequently its noise.

3) Embedment stage

The embedment stage is dedicated to the implementation of the watermarking process by applying the LSB method. However, this application is not performed successively, as the pixels of the watermark image are not embedded in the pixels of the cover image sequentially. Instead, it is suggested that the pixels of every watermark image's cluster are embedded in its nearest cover image's cluster. The purpose of that is to increase the security and imperceptibility levels of the watermarking yields. Furthermore, every pixel in a cluster of the watermark image is hidden by changing its first four bits with the last four bits of a pixel in the nearest cluster that belongs to the cover image, while the last four bits of the next cover image's pixel are substituted with the last four ones of that watermark image pixel, and so on. For example, assuming that the value of a pixel (that should be embedded) of a cluster which belongs to the watermark image is:

$$[1, 1, 1, 1, 0, 0, 0, 0]$$

whereas, the values of the two successive pixels in a cover image's cluster that is the closest to watermark image's cluster are:

$$[1, 0, 0, 1, 1, 0, 0, 1]$$

$$[1, 0, 1, 0, 1, 0, 1, 0]$$

After the embedding process, the above two pixels will have the following new embedded values:

$$[1, 0, 0, 1, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}]$$

$$[1, 0, 1, 0, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]$$

The above embedding manner helps to decrease both the visible changes in the cover image and its required size to embed all the pixels of the watermark image.

The procedure below illustrates the main steps for this stage:

For each cluster wl_J, $\forall J = 1, \dots, k$, in the watermark image:

1) Identify the closest cluster cl_j, $\forall j = 1, \dots, k$, to the cluster wl_J:

$$|M_D(wl_J, cl_j)| < |M_D(wl_J, cl_l)|, \forall l = 1, \dots, k, l \neq j$$

where $M_D(wl_J, cl_j)$ is the Manhattan distance between cluster wl_J and cluster cl_j.
2) Convert the decimal value of each pixel in both cluster wl_J and its closest cluster cl_j into its 8-bits binary equivalent, and then divide every obtained 8-bits into two halves.
3) Embed each pixel that belongs to cluster wl_J in two successive pixels belonging to cluster cl_j (the closest cluster to wl_J), such that the first half of a pixel that should be embedded is set to the last half of the first two successive pixels, while the last half of the second two successive pixels is filled with the last halve of a pixel that should be embedded.
4) Label each cover image's pixel that embeds secret bits with *ture* value by constructing a Boolean function called *hide*. This function takes two integer numbers representing a pixel's coordinate, and a colour space of an image that includes this pixel. The *hide* function shows whether a pixel with a given coordinate embeds confidential data or not. The *hide* function will be successfully used in the extraction phase.

4) Assessment stage

This stage aims at identifying the best chromosome procured by k-means clustering. The best chromosome refers to the optimal chromosome that yields the desirable concealment results. Therefore, during this stage, the fitness of each chromosome is assessed by using the Peak Signal to Noise Rate (PSNR) function [21]. It is defined as:

$$PSNR = 10log_{10}\frac{max^2}{MSE(C, W)} \qquad (2)$$

where *max* is the maximum luminance value in the cover image *C*, *W* is the watermarked image, and *mse* is the *Mean Square Error* between these images. The *MSE* is given by the following equation:

$$MSE(C, W) = \frac{1}{h \times w}\sum_{i=1}^{h}\sum_{j=1}^{w}(c_{ij} - w_{ij})^2 \qquad (3)$$

The calculated value of the above function is awarded to each chromosome to identify how fit this chromosome is.

5) Generation stage

During this stage, the two normal phases (crossover and mutation, in the genetic algorithm) are implemented to produce the next chromosomes generation.

The steps below briefly clarify the generation stage, as they are repeated until the optimal generation is attained, or until the maximum number of the produced generations is reached:

1) A pair of chromosomes is chosen from the current generation, so that this pair has the largest PSNR scores.
2) A crossover index on the chosen pair of chromosomes is randomly selected to produce new two chromosomes by swapping the bits of the chosen chromosomes between themselves until reaching the crossover index.
3) To preserve heterogeneity within a single generation, one or more bits of the newly produced chromosomes are randomly selected to swap their values within chromosomes.

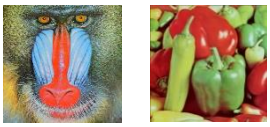
B. *Extracting Phase*

This phase is dedicated to extracting the watermark bits based on the defined *hide* function. The required steps for this phase can be outlined as follows:

1) Let $D_{h\times w}$ be a watermarked image of size $h \times w$.
2) The $WD_{h\times w}$ is split into three component colour matrices: $RD_{h\times w}$, $GD_{h\times w}$, and $BD_{h\times w}$.
3) For each colour matrix, the following is done:
   a) Each pixel belonging to the current colour matrix is converted to its binary format.
   b) The last four significant bits of a pixel are extracted, which were labeled with a *true* value by the *hide* function.
   c) The extracted bits are converted into their equivalent decimal format.
4) The three colour matrices are unified to yield the watermark image.

V. Experimental Results and Analysis

The proposed methodology is experimentally assessed using a set of 24-bit colour images, which are widely used in literature for image watermarking. The used set includes four host images: Baboon, Pepper, Toco Toucan, and Lichtenstein, each with a size of 256×256, and one watermark image of size 64×64, as shown in Fig. 3. Based on this set, the experimental analysis is conducted by implementing three tests, including the imperceptibility test, attacks test, and robustness test. All these empirical tests are implemented by using MATLAB software version 7.9.0 for Windows 10 operating system. In addition, the proposed methodology is compared with the k-means based LSB method to show how the genetic algorithm helps in enhancing the imperceptibility results. The subsections below illustrate the performed tests and their comparison.



(a) (b)

(c) (d)

(e)

Figure 3. The images' set: (a) baboon (b) pepper (c) toco toucan (d) lichtenstein (e) watermark.

A. *Imperceptibility Test*

To assess the imperceptibility of the watermarked image that is obtained through the proposed methodology, two metrics are utilized: the Peak Signal to Noise Rate (PSNR) and the Structural Similarity Index Measure (SSIM). PSNR (its equation is previously stated) is usually used to assess the quality of the watermarked image. As a result, the higher the PSNR value, the better imperceptibility of the watermarked image, and vice versa. Different from the PSNR metric, SSIM is developed to measure the similarity between the cover image and the watermarked image in terms of the luminance of each image, and the contrast of each image and the structure. It is calculated using the equation below:

$$SSIM = \frac{((2\mu_C\mu_W + \alpha)(2\sigma_{CW} + \beta))}{((\mu_C^2 + \mu_W^2 + \alpha)(\sigma_C^2 + \sigma_W^2 + \beta))} \quad (4)$$

where μ_C and μ_W are the mean of *C* and *W*, respectively; σ_C and σ_W are the variances of *C* and *W* respectively; σ_{CW} is the covariance of *C* and *W*, and α, β are constant values used to prevent null denominator. SSIM ranges between -1 and 1, whereby values closer to 1 have a greater similarity and 1 indicates that the two images are identical.

Table I shows the PSNR and the SSIM values for the used set images after applying the concealment phase. These values are calculated for the watermarked image with a different number of clusters varying from 30 to 70. From this table, it can be deduced that the proposed methodology attains sufficient imperceptibility results, as the PSNR values reach 45 dB and the SSIM values exceed 0.98 when the number of clusters increases.

TABLE I. Results of the Imperceptibility Test in Terms of PSNR and SSIM

Image	No. of Clusters=30		No. of Clusters=70	
	PSNR	SSIM	PSNR	SSIM
Baboon	47.5062	0.999694	47.6706	0.999747
Pepper	47.1643	0.999483	47.2788	0.999503
Toco Toucan	47.4292	0.998677	47.6274	0.998849
Lichtenstein	48.0915	0.999291	48.4546	0.999327

B. *Security Test*

Through this test, it is demonstrated how the proposed methodology can withstand deliberate attacks, including cropping, noise, histogram, filtering, and JPEG compression attacks. These attacks are directed towards a twofold aim: to verify the robustness of the proposed methodology, and to analyze how the k-means affects the

watermarking results. The number of clusters is equal to 30 and 70. The effects of the launched attacks are illustrated below.

1) Cropping attack: This attack implies cropping a certain portion of the watermarked image. This attack is launched by performing two experiments: the first one includes cropping half of the watermarked image, while the second experiment is carried out by cutting a quarter of the image. The attacked watermarked image of the first and second experiments are shown in Fig. 4(a) and Fig. 4(c), respectively, and their extracted watermark images are shown in Fig. 4(b) and Fig. 4(d). From the last two figures, it can be concluded that the extraction activity has sufficiently succeeded, as the proposed methodology applies the clustering process that leads to the distribution of the watermark pixels at different clusters (regions) of the cover image.

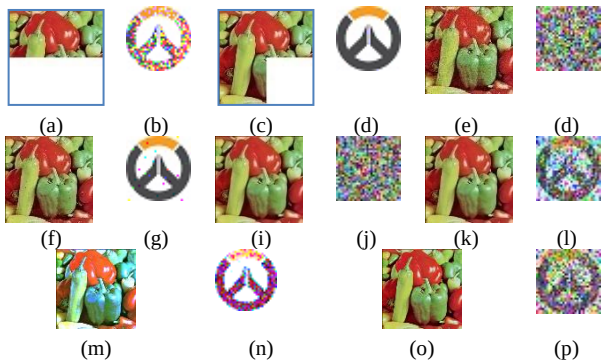


Figure 4. Pepper watermarked image that was attacked by: **(a)** Cropping ½ of watermarked image, **(c)** Cropping ¼ of watermarked image, **(e)** Gaussian noise variance (v)=0.01, **(g)** Salt & pepper noise variance (v)=0.01, (i) Average filter 3×3, **(k)** Median filter 3×3, **(m)** Histogram equalization, **(o)** Compression attack with ratio=3 under the JPEG 2000. Each watermarked image is with its corresponding extracted watermark at **(b), (d), (f), (h), (j), (l), (n)** and **(p).**

2) Adding noise: The watermarked image is further examined by launching the 'Gaussian' noise attack and the 'salt-and-pepper' attack at it, see Fig. 4(e) and Fig. 4(g). In general, these attacks have undesirable effects on the watermarked image appearance. However, only a little information about the extracted watermarks is lost in the proposed methodology, as shown in Fig. 4(f) and Fig. 4(h).

3) Median and average filters attack: The watermarked image is also tested by launching low-pass-filter (average filter) and median-filter attacks with a window size of 3×3, as shown in Fig. 4(i) and Fig. 4(k), illustrating the results of attacking the watermarked images. The result shows that there is less distortion in the watermark extracted from the median filter-attacked image than from the average-attacked one.

4) Histogram equalization attack: The watermarked image is tested by modifying its histogram, as in Fig. 4(m). The experimental result indicates that after applying this attack, little information about the extracted watermark is lost.

5) JPEG compression attack: This attack is usually considered critical as it leads to the loss of most of the least significant bits of the watermarked image. However, the method proposed in this study can resist this attack as it distributes the embedding bits of the watermark image relying on the k-means clustering. The attacked watermarked image and its corresponding extracted watermark are shown in Fig. 4(o) and Fig. 4(p), respectively.

Under all of the aforementioned attacks, the imperceptibility of the proposed methodology is analyzed in terms of PSNR and SSIM values, see Table II. From this table, it can be stated that this methodology attains the highest robustness against salt and pepper and cropping attacks. With respect to the JPEG compression attack, Table II lists the PSNR and SSIM values with different quality factors ranging from 10 to 80 and certain compression ratio, including 3, 5, 10, 15, 20, and 25. The results show that the resistance to JPEG attack increases when the quality factor is increased and the compression ratio is relatively small. In Table II, the abbreviations QF and CR stand for Quality Factor and Compression Ratio, respectively.

TABLE II. PSNR AND SSIM VALUES OF THE PEPPER WATERMARKED IMAGE AFTER ATTACKING IT

Attack	PSNR		SSIM	
	No. of clusters= 30	No. of clusters= 70	No. of clusters= 30	No. of clusters= 70
Median filter 3×3	30.584	30.5884	0.983573	0.98358
Average filter 3×3	27.449	27.4499	0.969619	0.969636
Salt & pepper noise	24.6022	24.9653	0.944832	0.939409
Gaussian noise	20.3274	20.3075	0.838836	0.837358
Cropping ¼	9.6537	9.6537	0.750451	0.750473
Cropping ½	6.8162	6.8162	0.50922	0.509214
Histogram equalization	13.9342	13.9354	0.582467	0.582218
JPEG QF=10	24.3591	24.3595	0.933763	0.933744
JPEG QF=20	26.6631	26.6592	0.95956	0.959522
JPEG QF=30	27.8688	27.8774	0.968692	0.968679
JPEG QF=40	28.6594	28.7584	0.976377	0.977401
JPEG QF=50	29.3594	29.3603	0.977377	0.977588
JPEG QF=60	30.0139	30.0101	0.980403	0.980358
JPEG QF=70	30.8886	30.8944	0.983715	0.983703
JPEG QF=80	32.2263	32.2322	0.987707	0.987695
JPEG CR=3	45.2592	45.2959	0.999244	0.999255
JPEG CR=5	42.3322	42.4078	0.998603	0.998636
JPEG CR=10	36.8377	36.8661	0.995374	0.995348
JPEG CR=15	34.3781	34.3971	0.991939	0.991882
JPEG CR=20	32.6744	32.6744	0.988429	0.988427
JPEG CR=25	31.3505	31.3388	0.984782	0.984762

C. Robustness Test

The robustness of the extraction process is inspected using the Normalized Cross Correlation (NCC) metric [21]. It is employed as a similarity measurement between the original inserted watermark and the extracted one. The NCC takes a value between 0 and 1, and it is calculated using the following formula:

$$NCC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} O_{i,j}E_{i,j}}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} O_{i,j}^2}\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} E_{i,j}^2}} \quad (5)$$

where O and E are the original and the extracted watermark, respectively; M and N denote the image dimensions. When the value of the NCC is equal to 1 then both the inserted watermark and the extracted one are identical. Meanwhile, when its value is greater than 0.75, then the output of the extraction process is considered reasonable.

Using the above equation, the NCC values for the used images set are calculated. However, due to the space restriction, only the NCC values for the pepper image sample are displayed in Table III. The observed NCC values in this table are 1 when there is no attack of the pepper watermarked image under a different number of clusters, whereas they tend to be closer to 1 in case the watermarked image undergoes a malicious attack.

TABLE III. NCC VALUES WITH AND WITHOUT ATTACKS OF THE PEPPER IMAGE SAMPLE

Attack	No. of clusters=30	No. of clusters=70
No attack	1	1
Median filter 3×3	0.921773	0.978625
Average filter 3×3	0.852918	0.857082
Salt & pepper noise	0.996351	0.99705
Gaussian noise	0.794904	0.879603
Cropping ¼	0.999998	0.999998
Cropping ½	0.937777	0.950107
Histogram equalization	0.8819	0.936955
JPEG QF=10	0.866727	0.890112
JPEG QF=20	0.849149	0.863089
JPEG QF=30	0.849149	0.87084
JPEG QF=40	0.847425	0.878903
JPEG QF=50	0.847425	0.868903
JPEG QF=60	0.842744	0.870531
JPEG QF=70	0.854643	0.873467
JPEG QF=80	0.863129	0.8819
JPEG CR=3	0.817236	0.89659
JPEG CR=5	0.84555	0.897447
JPEG CR=10	0.873424	0.88676
JPEG CR=15	0.872847	0.885192
JPEG CR=20	0.868243	0.888342
JPEG CR=25	0.872309	0.881647

D. Comparison

To demonstrate the effectiveness of this methodology, its imperceptibility results are compared with those of the k-means-based LSB method. Fig. 5 and Fig. 6 show the PSNR values for both methods using 30 clusters and 70 clusters, respectively. These values are clear evident that the imperceptibility performance of this methodology is higher than the k-means-based LSB method by virtue of the integration with the genetic algorithm.

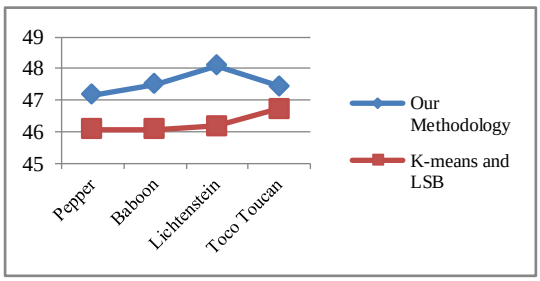


Figure 5. Comparison of PSNR values between the proposed methodology and k-means-based LSB method using 30 clusters.

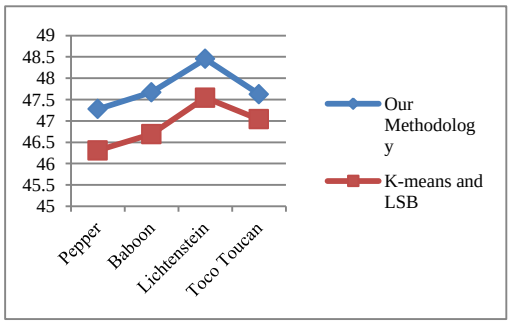


Figure 6. Comparison of PSNR values between the proposed methodology and k-means based LSB method using 70 clusters.

The SSIM values related to this methodology are also compared with those of k-means based LSB method using 30 and 70 clusters, see Fig. 7 and Fig. 8 From these figures, it can be stated that the structural similarity of the proposed methodology is better that that of the k-means based LSB method.

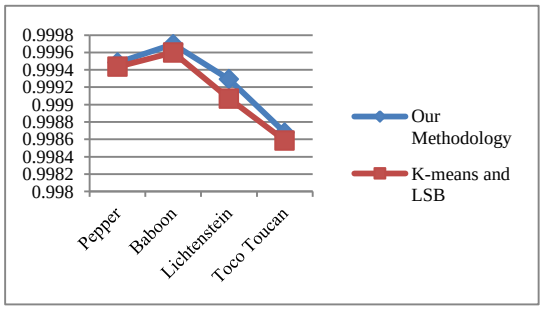


Figure 7. Comparison of SSIM values between the proposed methodology and k-means based LSB method using 30 clusters.

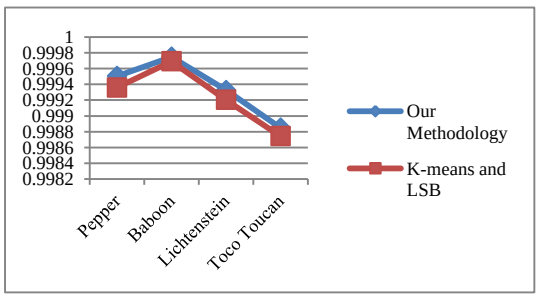


Figure 8. Comparison of PSNR values between the proposed methodology and k-means based LSB method using 30 clusters.

VI. CONCLUSION

In this paper, a genetic k-means clustering is proposed for colour image watermarking. The Genetic Algorithm is collaborated with the k-means clustering technique to produce the optimum cluster centroids for every colour component of an image. In addition, the convergence to the best cluster centroids is accelerated. On the other hand, the k-means clustering technique helps to increase the imperceptibility level, as the embedment of the watermark pixels is based on the nearest clusters having pixels with similar or closer values to those of the watermark pixels. Furthermore, as the bits of pixels belonging to the watermark image are embedded in different pixels of the

cover image, this raises the robustness level of the watermark to inhibit noise.

Our future directions include the application of the proposed methodology using different hiding and clustering methods, such as Pseudorandom LSB and Fuzzy C-means.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The research is conducted under the supervision of Farah Al-Shareefi, Zainab Falah Hassan analyzed the data, and Hadeel Qasem Gheni and Farah Al-Shareefi wrote the paper. All authors approved the final version.

REFERENCES

[1] A. Mohanarathinam, S. Kamalraj, G. K. D. Prasanna, V. Renjith, *et al*., "Digital watermarking techniques for image security: A review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 8, pp. 3221–3229, 2020. doi: 10.3390/info11020110

[2] G. Suresh, V. L. Narla, D. P. Gangwar, and A. K. Sahu, "False-positive-free SVD based audio watermarking with integer wavelet transform," *Circuits, Systems, and Signal Processing*, pp. 1–26, 2022. doi: 10.1007/s00034-022-02023-5

[3] M. Kaczyński and Z. Piotrowski, "High-quality video watermarking based on deep neural networks and adjustable subsquares properties algorithm," *Sensors*, vol. 22, no. 14, p. 5376, 2022. doi: 10.3390/s22145376

[4] N. Jing, Q. Liu, and V. Sugumaran, "A blockchain-based code copyright management system," *Information Processing & Management*, vol. 58, no. 3, 102518, 2021. doi: 10.1016/j.ipm.2021.102518

[5] R. Sinhal, D. K. Jain, and I. A. Ansari, "Machine learning based blind colour image watermarking scheme for copyright protection," *Pattern Recognition Letters*, vol. 145, pp. 171–177, 2021. doi: 10.1016/j.patrec.2021.02.011

[6] M. Begum and M.S. Uddin, "Digital image watermarking techniques: A review," *Information*, vol. 11, no. 2, p. 110, 2022.

[7] B. S. Duran and P. L. Odell, *Cluster Analysis: A Survey*, Springer Science & Business Media, 2013.

[8] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021. doi: 10.1007/s11042-020-10139-6

[9] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. 1st International Conf. On Image Processing*, vol. 2, pp. 86–90, IEEE, 1994. doi: 10.1109/ICIP.1994.413536

[10] G.-J. Lee, E.-J. Yoon, and K.-Y. Yoo, "A new LSB based digital watermarking scheme with random mapping function," in *Proc. 2008 International Symposium on Ubiquitous Multimedia Computing*, 2008, pp. 130–134. doi: 10.1109/UMC.2008.33

[11] M. T. B. Othman, "Digital image watermarking based on image clustering," in *Proc. the 3rd International Conf. on Circuits, Systems, Communications, Computers and Applications (CSCCA'14)*, Florence, Italy, 2014.

[12] R. J. Essa, N. A. Abdulah, and R. D. Al-Dabbagh, "Steganography technique using genetic algorithm," *Iraqi Journal of Science*, pp. 1312–1325, 2018.

[13] W.-C. Yang, C.-Y. Wen, and C.-H. Chen, "Applying public-key watermarking techniques in forensic imaging to preserve the authenticity of the evidence," in *Proc. International Conf. on Intelligence and Security Informatics*, Springer, 2008, pp. 278–287. doi: 10.1007/978-3-540-69304-8

[14] R. Suganya and D. R. Kanagavalli, "Tamper detection using watermarking scheme and k-mean clustering for bio medical images," *International Journal for Modern Trends in Science and Technology*, vol. 2, pp. 180–185, 2016.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975. doi: 10.1137/1018105

[16] J. L. R. Filho, P. C. Treleaven, and C. Alippi, "Genetic-algorithm programming environments," *Computer*, vol. 27, no. 6, pp. 28–43, 1994. doi: 10.1109/2.294850

[17] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 2009.

[18] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Inc., 1988.

[19] F. Szabo, *The Linear Algebra Survival Guide: Illustrated with Mathematica*, Academic Press, 2015.

[20] M. Mohibullah, M. Z. Hossain, and M. Hasan, "Comparison of Euclidean distance function and Manhattan distance function using k-mediods," *International Journal of Computer Science and Information Security*, vol. 13, no. 10, p. 61, 2015.

[21] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Transactions on Communications*, vol. 43, no. 12, pp. 2959–2965, 1995. doi: 10.1109/26.477498

Zainab Falah Hassan was born in Iraq / Babylon in 1984, received a bachelor's degree in computer science from the university of Babylon, faculty of science for women, computer department, Iraq, in 2006, and a master's degree in artificial intelligence from the university of Babylon, faculty of science, Iraq, in 2013. She is currently a lecturer at the university of Babylon, faculty of science for women, computer department. Her current research interests include artificial intelligence, image processing, and machine learning.

Farah Al-Shareefi was born in Iraq / Babylon in 1982, received a bachelor's degree in computer science from the university of Babylon, faculty of science, computer department, Iraq, in 2004, and a master's degree in face recognition from the university of Babylon, faculty of science, computer department, Iraq, in 2011, and a Ph.D. degree in computer science from the university of Liverpool, Liverpool, L69 3BX, United Kingdom in 2019. She is currently a lecturer at the university of Babylon, college of science for women, computer department, Her current research interests include security, safety and security analysis, system and software engineering, formal methods, and image processing.

Hadeel Qasem Gheni was born in Iraq / Babylon in 1984, received a bachelor's degree in computer science from the university of Babylon, faculty of science for women, computer department, Iraq, in 2006, and a master's degree in artificial intelligence from the university of Babylon, faculty of information technology, software department, Iraq, in 2016. She is currently a lecturer at the university of Babylon, faculty of science for women, computer department, Iraq. Her current research interests include artificial intelligence, machine learning, and data mining.