IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 May 2023, accepted 11 June 2023, date of publication 16 June 2023, date of current version 21 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3286865

RESEARCH ARTICLE

Protecting the Distribution of Color Images via Inverse Colorization, Visible-Imperceptible Watermarking and Reversible Data Hiding

EDUARDO FRAGOSO-NAVARRO[1], (Member, IEEE), FRANCISCO GARCIA-UGALDE[1], AND MANUEL CEDILLO-HERNANDEZ[2]

[1]Facultad de Ingenieria, Universidad Nacional Autónoma de México (UNAM), Coyoacán, Mexico City 04510, Mexico
[2]Escuela Superior de Ingenieria Mecanica y Electrica, Instituto Politécnico Nacional (IPN), Coyoacán, Mexico City 04430, Mexico

Corresponding author: Manuel Cedillo-Hernandez (mcedilloh@ipn.mx)

This work was supported in part by Universidad Nacional Autónoma de México (UNAM) under the Dirección General de Asuntos del Personal Académico (DGAPA) Postdoctoral Scholarship Program and Research Project under Grant PAPIIT-IT100123, in part by Instituto Politécnico Nacional (IPN), and in part by Consejo Nacional de Ciencia y Tecnología (CONACYT).

ABSTRACT Invertible color-to-gray algorithms are good alternatives for protecting color images in storage, transmission, or limited-access environments. These approaches hide color information in a gray-scale version of an image. Thus, illegal recovery of the original color becomes a challenging task, and only authorized users can restore the colorized image. However, although the color is protected, the original structure of the image remains vulnerable to illegal usage. In this paper, we present a reversible distortion of the protected gray-scale image to make illegal reconstruction more difficult. In addition, we preserve the general visibility of the original content. Furthermore, a visible imperceptible watermark is embedded to protect ownership of the colorized image in public access. The watermark remains imperceptible in the colorized image and reveals the logo of the owner when the image is protected. We propose different levels of distortions obtaining mean PSNR qualities between 14.74 dB and 32.92 dB with respect to a reference gray-scale image. Furthermore, the mean PSNR quality of the colorized images remains between 38.87 dB and 41.55 dB.

INDEX TERMS Colorization, contrast enhancement, image distribution, image protection, invertible color-to-gray, reversible data hiding, visible-imperceptible watermark.

I. INTRODUCTION

The widespread use of image capture devices, social media, and online multimedia services has increased the generation, transmission, and publication of color images. One of the main concerns regarding stolen or published images is the unauthorized use of their content such that security systems may be required. In this regard, some solutions are proposed by invertible color-to-gray algorithms. These approaches protect a color image by converting it into a gray-scale version that contains the color information hidden. Thus, only authorized users can extract the color information and obtain a colorized version of the image. In this manner, the owner can share a gray-scale preview of the image content, and users can decide to ask for the authorized key to completely reconstruct the image. The objective is that the illegal attempt to recover the original color from the protected image becomes extremely challenging.

Invertible color-to-gray methods can be divided into three main categories: subband embedding (SE)-based methods [1], [2], [3], [4], [5], vector quantization (VQ) schemes [6], [7], [8], [9], [10], [11], [12], [13], [14], and convolutional neural network (CNN)/deep learning (DL) approaches [15], [16], [17], [18]. However, some new algorithms [19] may not fit the previous classifications. SE schemes [1], [2], [3], [4] hide a down-sampled version of the chromas in a subband transform of the luminance. The frequencies of the subband to hide the color are selected

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

to avoid reducing the quality of the luminance. However, the loss of some high-frequencies and the reduction of the chromaticity lead to blurring of the color and the luminance planes. VQ schemes [6], [7], [8], [9], [10], [11], [12], [13], [15], [16], [17], [18] offer a different approach by generating a color palette and an index image. The former is also a gray-scale version image, and its indexes point to the color of the palette that corresponds to each region. Therefore, the colorization is a table look-up process, which is highly efficient. However, the hiding process consists of embedding the color palette into the index image, that may cause index information loss and a distorted colorization. Finally, CNN/DL approaches [15], [16], [17], [18] uses a big dataset to train a CNN. The network learns to generate a gray-scale image with the color information both encoded and hidden. Most of the CNN schemes generates a visual encoding of the color, perceived as some pattern distortions into the output image. However, some schemes such as [17] have reduced considerable the visual distortions.

Invertible color-to-gray algorithms seek the highest possible qualities for the gray-scale and colorized images, simultaneously. The gray-scale image protection consists of the difficulty of the illegal color extraction and colorization. However, although color is not available, the structural information is vulnerable to illegal usage since it is visible in the protected image. In this regard, cryptography may be used to eliminate the visual structure. Cryptography is one of the most extreme schemes for content protection because it can generate useless images with no visual meaning. However, many applications require the protected image to partially show the content of the original information.

Therefore, we propose a new invertible color-to-gray method that yields distorted, but recognizable gray-scale images. The key idea is to increase security and offer intermediate protection between cryptography and the previous invertible color-to-gray approaches. Thus, illegal restoration becomes more difficult because it is necessary to recover both color and structure. Therefore, we propose two modes based on the desired security level. The balanced mode increases the contrast of the gray-scale image using a reversible data hiding with contrast enhancement (RDH-CE) algorithm [20]. This makes it more difficult to recover the original luminance illegally because the contrast of the protected image is modified. The distorted mode further alters the structural quality of the protected image, with uneven contrast luminance and block effects. This is achieved by increasing the hidden load in image blocks. Finally, an optimization process [21] adjusts the algorithm parameters to obtain balanced and distorted results. The decolorization-colorization algorithm is an improvement of the proposed scheme in [22]. Additionally, to protect the ownership of the color image, we applied a visible imperceptible watermark. A watermarked image contains the logo of the owner, which is embedded in the original luminance of the image. The watermark is embedded with weak strength, so the logo is not visible at first sight in the colorized image. The logo is then visually revealed with the contrast increment when hiding the color information. Namely, the imperceptible logo remains when the color image has public access, and the owner can be identified anytime the contrast is incremented.

The contributions of this paper are summarized as follows:

1. We propose a new invertible color-to-gray method that distorts a protected gray-scale image to hinder the illegal reconstruction of the original image structure.
2. The present study offers two levels of distortion that can be selected according to the application.
3. We achieve to integrate a visible-imperceptible watermark for authentication with the invertible color-to-gray protection.
4. Image distortion increases image security in storage, transmission, and limited access, and the watermark protects ownership in public access.

The remainder of this paper is organized as follows. A summary of the main related works are presented in section II. The expected results, general descriptions, and detailed techniques are presented in section III. Section IV presents and discusses the results. A discussion is presented in Section V. Finally, the conclusions and future work are presented in Section VI.

II. RELATED WORKS

A. INVERTIBLE COLOR-TO-GRAY COLORIZATION

Decolorization-colorization and reversible data hiding are the two main processes of invertible color-to-gray algorithms. Firstly, decolorization separates both luminance and color information of a color image. Then, the color may be coded or compressed to be subsequently hidden into the luminance. The result is a grey-scale image which is the protected version of the content. Finally, in colorization stage the color is extracted, decoded, and restored to obtain a colorized image. The main objective of previous approaches is to obtain the best quality possible for both gray-scale and colorized images.

1) SE-BASED METHODS

SE-based methods substitute subband coefficients in the wavelet transform of the luminance by down-sampled chromaticity. In [1], due to the high frequency alteration in the wavelet transform, some interference pattern appears in gray-scale image. However, this pattern makes the color recovering robust to print-scan attacks. On the other hand, the colorized image suffers from pattern visualization and low color saturation, even without attack. Therefore, some post-processing helps to reduce the high undesirable frequencies and lack of saturation. Reference [2] describes a redundant embedding process based on noise analysis to increase the quality of the protected image. The proposal in [5] selects the subbands with the least energy. Therefore, it achieves a better gray-scale image quality. The results show a less noticeable pattern in the protected image, and a better color saturation in the colorized image. In [3], the limits of the color values are hidden and used to recover the original

color range. Therefore, the saturation improves considerably. The embedding in [4] selectively scatters the color information in non-consecutive subband coefficients. This way, the gray-scale and color quality increases. Additionally, a random generator key for distributing the information provides additional security against illegal extraction.

One of the main characteristics of SE-based approaches is the pattern interference in the gray-scale image. Most of the pattern is then transferred to the colorized image, decreasing its quality. Additionally, the down-sample process in the chromaticity causes a lack of saturation. Although most recent works improve pattern and saturation, the achieved quality may not be enough for more demanding applications.

2) VQ-BASED METHODS

VQ-based approaches generate a color palette and an index image. The palette contains a reduced number of colors that best represent the original colors. On the other hand, each index in the image is the location of the palette with the most adequate color for the corresponding pixel. Therefore, the efficient colorization process comprises a simple table search. The algorithm makes the index image being a gray-scale version by ordering the color palette such that each index keeps the same visual proportion to the luminance values. Additionally, since the palette is necessary in colorization, its color information is hidden in the protected image. Proposal in [14] calculates a palette with 256 colors using fuzzy c-mean approach. The palette is then hidden using LSB substitution. The results show a gray-scale image with higher contrast than the luminance plane. Authors in [8] improves qualities of protected and colorized images by using a fast quantization approach with K-means classifier for palette generation, obtaining better quality in the colorized image. Then, the color information is compressed to increase the quality of the gray-scale image after LSB hiding. However, the high contrast effect in the protected image remains. Subsequent schemes take this contrast as an undesirable effect and propose solutions to increase the similarity between the gray-scale versions of the image and the luminance. Authors in [7], proposes an energy function to generate a palette that disappears the contrast effect in the luminance at a cost of some false edges in the colorized image. Authors in [6], reduces the false edges in [7] with a k-means clustering with lightness constrains. The scheme in [9] substantially improves colorized image quality by increasing the palette colors to 512. Since the hiding payload increases, they replace the LSB substitution by a RDH approach. RDH scheme recovers the original index image reducing the possible incorrect indexing and increasing the color quality. In [11], error diffusion method distributes color in a halftoning effect. Therefore, the false borders caused by the limited color representation is replaced by points pattern, which may be more visually pleasing. Most schemes do not include an additional security method, however, authors in [12] propose to embed a fragile watermark for tamper detection. The watermark process is capable of colorize the parts of the gray-scale image that were illegally altered. The scheme without watermark improves the quality of previous schemes, and the watermark-based approach does not affect considerably de quality of the resulting images. Then, proposal [13] improves the results in [12] by using a RDH method to hide the color information, instead of the conventional LSB approach. Unlike [9], the RDH algorithm is applied directly to the original luminance instead to in the index image. This method increases the quality of the colorized image since the original luminance and the exact indexes are recovered.

The last VQ-based algorithms have achieved good quantitative qualities for the gray-scale and colorized images. However, the decrease of the number of colors to 256 or 512 is always a limitation resulting in false borders or halftoning patterns. Despite these limitations, VQ-based algorithms continue improving and we consider three important contributions that may not have been sufficiently exploited. Firstly, the high contrast of the gray-scale image achieved in [8] and [14] are not considered a benefit. However, this may be an additional security approach since the illegal reconstruction of the image becomes more difficult. Secondly, we believe that additional security approaches as the proposed in [12] are necessary to cover security loopholes. The implementation of such approaches may not be easy since they should not affect considerable the quality of the images. Finally, we consider that the RDH implementation in [9] and [13] has been a right choice to recover the original information and improve colorization. In the present proposal, we take into consideration previous three benefits of VQ-based approaches: a contrast increment, an extra security approach for ownership authentication, and the implementation of an RDH algorithm.

3) CNN-BASED METHODS

In CNN-based schemes, a color image feeds the trained network. The CNN generates a gray-scale image that contains the encoded color information. Authors in [15] implemented an encoder and decoder based on a vanilla U-net. The encoder yields a gray-scale image with the encoded color displayed in the form of subtle patterns. Then, the decoder converts those patterns into color. The loss function comprises terms that contributes to invertibility, gray-scale conformity and quantization values of the color. In [16], a dual features ensemble network is proposed. Additionally, different loss functions are tested to know the contribution of each term. However, we can see that the loss function can prioritize one of both, the colorized or the gray-scale images. Therefore, incrementing the quality of one decreases the quality of the other. However, the most equilibrated option is selected, and further analysis is not achieved. Reference [17] trains a reversible network using the wavelet domain. The approach considerable reduces the codification lattice which become visually negligible in most of the gray-scale image. Additionally, the quality of the colorized images is considerably improved. Authors in [18] propose a generative adversarial network to make the colorization process robust to JPEG compression.

CNN approaches have shown a promising solution to invertible color-to-gray problems. However, some patterns may be found, and color blurriness are generated in some details, such as edges. Since coding and hiding are included in a unique step forward process of the network, the training attempts to find the best equilibrium between decolorization and colorization. Improvements are achieved by adjusting the network structure, the conditions in the loss function, and the training database. However, we believe that a modular structure such as VQ-based and SE-based approaches in which colorizing and hiding have separate processes, may facilitate the predictability of the results according to adjustments.

4) ADDITIONAL APPROACH

Some proposals may not fall within previous classification, but they can yield good results. One of these approaches is proposed in [19]. Firstly, the luminance and chrominance are separated. Then, the chrominance planes are compressed by JPEG2000 algorithm and hidden into the luminance. Two hiding methods are tested, one RDH and the conventional LSB substitution. RDH degrades more the gray-scale image, but reversibility recovers the original luminance and yields a better colorized image. However, RDH method is partially discarded since does not have enough capacity to hide the color information in all cases. Therefore, LSB substitution is then recommended to hide the color information, obtaining better qualities than previous approaches.

B. COLORIZATION

In general, colorization algorithms are used in three main applications: image compression, image restoration, and invertible color-to-gray protection. Image compression approaches compress the color and the luminance separately to generate a lower file size. Then, the color is decompressed, and the luminance colorized. Colorization for image restoration is used to edit a color image or to colorize an old image with no color. Some algorithms automatically select the palette and colorize the image. In other schemes, the desired colors are defined in some representative pixels (RPs), and the color is then expanded according with some conditions. However, we consider that colorization schemes used in invertible color-to-gray approaches have been designed separately without considering the colorization approaches for compression and restoration.

In image compression and restoration, we can identify two general groups of colorization schemes [23]. Some colorization schemes are primarily based on classical image processing approaches, whereas others are based on CNNs. CNN methods have become dominant since 2015 but because they require large datasets, classical approaches are still of interest.

The first non-CNN colorization method using predefined colors in RPs was proposed in [24], for restoration. The colorization stage comprises an optimization that propagates each RP color with the support of luminance values. The proposal in [24] have been modified for compression applications [25], [26]. However, some characteristics of RP-based methods may hinder colorization. First, one RP chrominance value may not adequately represent the original values of all surrounding pixels. Second, one luminance value may correspond to more than one color, causing a mismatched colorization. Finally, propagation may cause low color adherence to image borders. The optimization compensates previous problems to obtain good quality with the least possible color information. However, we consider that the method described in [22] has not been fully explored and can avoid the problems of methods based on RPs. Instead of recovering a propagated color, the proposed method [22] recovers a color signal that is similar to the original one at each superpixel [27] segment. Owing to segment-based processing, this approach exhibits good adherence to superpixel borders. Therefore, the proposed colorization scheme was based on this approach. To improve the tradeoff between color adherence and compression, we designed an improved segmentation method. The proposed segmentation generates large segments for redundant and small segments for further detail.

III. PROPOSED SCHEME

The general purpose of the proposed algorithm is to hide the color information of image I in its luminance channel. The output is a gray-scale image Yc whose contrast is increased with respect to the original luminance. Subsequently, in the colorization process, the color information can be extracted from the modified luminance, and the original information can be recovered. Finally, the luminance is colorized using the extracted information to obtain a colorized image I' that is visually similar to the original image. Fig. 1 shows an example of this proposal. If we repeat the hiding process with I' as the input and apply the colorizing process to the contrasted luminance Yc, we can recover the same colorized image I'. Therefore, we can switch between the colorized image I' and the contrasted luminance Yc in a completely reversible manner.

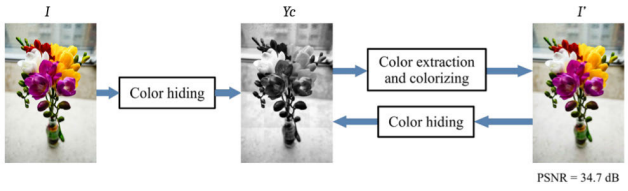


FIGURE 1. Reversibility process between Yc and I'. The color image is taken from COCO dataset [28].

In other words, a gray-scale version of the color image can contain all the information required for colorizing itself. This luminance is one channel in size and can be stored in a local device or cloud service for subsequent sharing. If the security of the storage medium is compromised and the image is illegally extracted, it cannot be used normally because it does not contain color.

To increase the level of protection, we propose two additional methods that mainly affect the contrasted luminance. These additional approaches give four security options, as shown in Fig. 2. The first approach decreases the structural quality of the contrasted luminance when hiding the color information, making unauthorized reconstruction more difficult. This technique can be achieved by defining the objective function of an optimization method, such that the luminance quality decreases. This function defines the trade-off between the quality of the colorized image and contrasted luminance. In the distorted mode, we define the function such that the colorized image has the best possible quality, whereas the contrasted luminance is distorted. In the balanced mode, the qualities of the colorized and contrasted images increase simultaneously. In the second approach, visible-imperceptible watermarks with the owner logo are embedded into the color image. The watermark in the colorized image is not visible at first sight and is visually revealed in the contrasted luminance, when the color information is hidden. Thus, the owner logo toggles between hidden and revealed when the image toggles between colorized and gray-scale, respectively. Thus, the copyright of the colorized image can be detected outside the storage device in case of unauthorized usage. Fig. 2. shows examples of the contrasted luminance for each possible solution of the proposed method. We can see in the second column the distortion effect, and a magnification is achieved in the second row to see that the watermark is reveled. On the other hand, Fig. 3 shows the colorized images corresponding to the four security options of Fig. 2. The objective is that four colorized images are qualitative and visually similar, with high quality. Therefore, it is important to notice that the visual and PSNR qualities of the colorized images are not considerably affected no matter the security mode. Furthermore, the watermark is hidden at first sight even with magnification.

In the following subsections, we first present a general description of our proposed algorithm. Then, we describe in detail the image segmentation, correction of segmentation, colorization process, proposed RDH-CE method, and optimization scheme implementation. Finally, a visible imperceptible watermark is described.

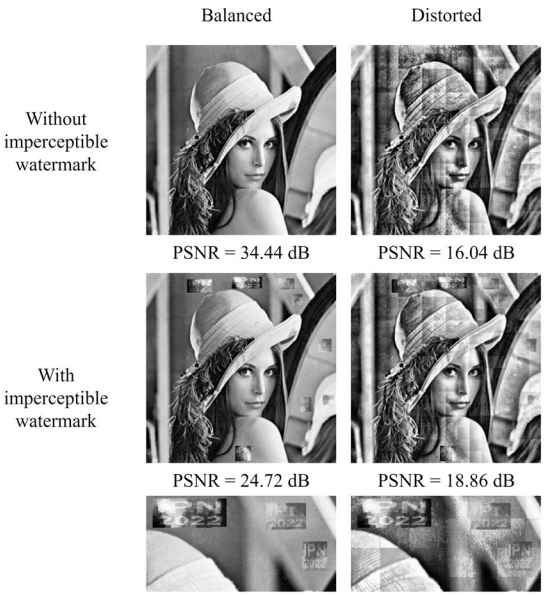


FIGURE 2. Contrasted luminance with color information hidden for the four security solutions.

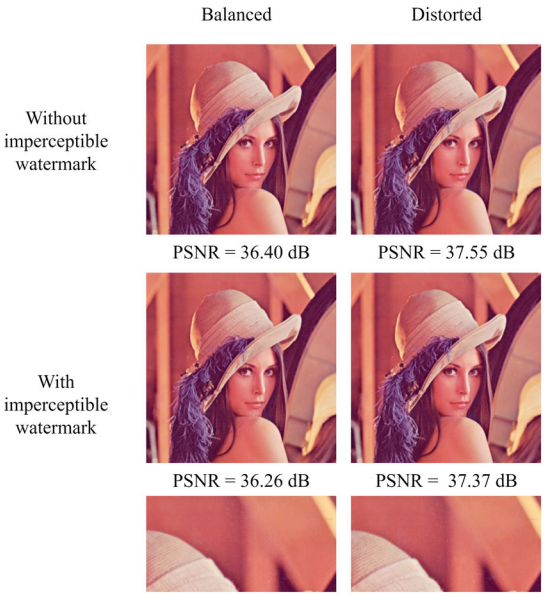


FIGURE 3. Colorized images for the four security solutions.

A. GENERAL DESCRIPTION

The basic decolorization and colorization processes are shown in Fig. 4(a) and Fig. 4(b), respectively. As shown in Fig. 4(a), the color image I in RGB color space is converted into the $YCoCg$ color space as follows:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{1}$$

The luminance Y is divided into segments $Segs = \{Seg_1, Seg_2, \ldots, Seg_N$ using a specific segmentation algorithm. Then, four coefficients per segment, two for Co and two for Cg, are calculated to approximate the chrominances Co and Cg using the luminance Y. The sets of coefficients are defined as $Ao = \{Ao_1, Ao_2, \ldots, Ao_N\}, Bo = \{Bo_1, Bo_2, \ldots, Bo_N$ for the Co channel and $Ag = \{Ag_1, Ag_2, \ldots, Ag_N\}, Bg = \{Bg_1, Bg_2, \ldots, Bg_N$ for the Cg channel, where each element in the set corresponds to each segment in $Segs$. Fig. 4(b) shows the colorizing process, where the luminance Y and the coefficients in Ao, Bo, Ag and Bg are used to obtain the approximated chrominances Co' and Cg', and obtain the colorized image I'.

The basic processes shown in Fig. 4 are used for the complete colorization in Fig. 5. As shown in Fig. 5(a), the color

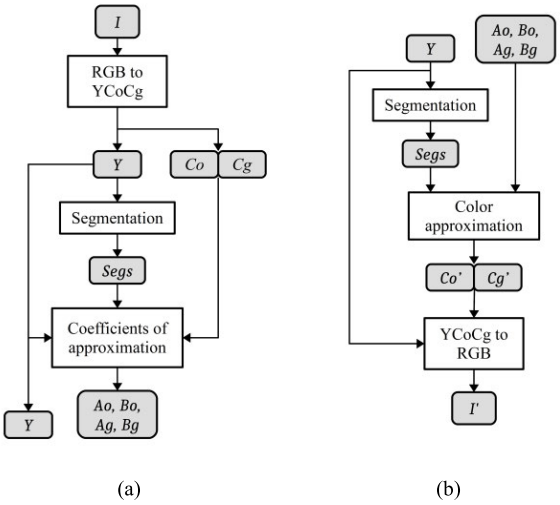


FIGURE 4. Basic (a) decolorization and (b) colorization processes.

image I is transformed into the color space $YCoCg$. Second, the luminance component Y is segmented in LSC superpixels [29] $S = \{S_1, S_2, \ldots, S_N\}$, and the number of segments is then reduced in cluster regions $C = \{C_1, C_2, \ldots, C_M\}$, where $M < N$, using the approach described in Section III.B. In our proposed colorizing method, we assume a high correlation between luminance and color. Nevertheless, this may not always occur; therefore, if we colorize each segment of luminance Y, the colors of some areas will not be recovered properly. To address this problem, an evaluation process is performed to find and correct faulty segmentation using color image I and the segments in S and C. The correction method decolorizes and colorizes the image according to the procedure shown in Fig. 4. First, colorization is achieved using cluster C segmentation, the colorized image is evaluated, and a new segmentation is proposed. The second evaluated segmentation includes the clusters together with the faulty superpixel areas, and the result returns superpixels that improved and superpixels that did not. The latter superpixels are then divided into two segments to create the final segmentation *Segs*. In addition, the correction algorithm returns three data points to recover the final segmentation without any evaluation. That is, the superpixel indexes S_B^{idx} that are faulty and improved in colorization, the indexes S_C^{idx} of the superpixels that are segmented into two parts, and the segmented pixels S_C^{pxls} of the segmentation applied to the set of superpixels indicated in S_C^{idx}. A detailed description of segmentation correction is provided in Section III.C. Once we obtain the final segmentation *Segs*, we calculate the coefficients Ao, Bo, Ag and Bg according to the colorizing method described in Section III.D. The coefficients are hidden in luminance Y together with S_C^{idx}, S_C^{idx} and S_C^{idx} using the RDH-CE embedding method described in Section III.E, obtaining the contrasted luminance Yc.

On the other hand, Fig. 5(b) shows a block diagram of the colorizing process. First, we applied the RDH-CE extraction

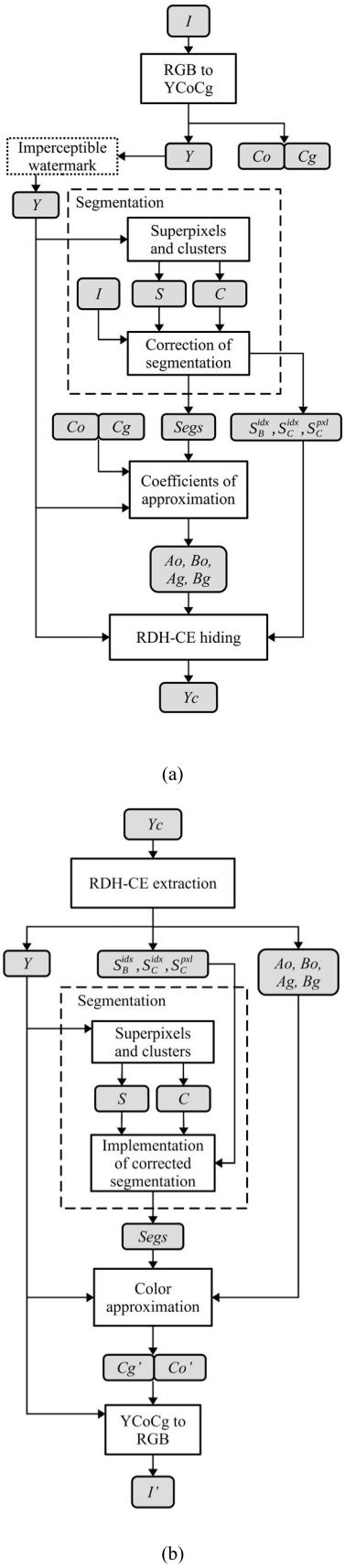


FIGURE 5. (a) Color hiding method. (b) Color extraction and colorizing.

algorithm to the luminance image Yc, recovering the original luminance Y, the superpixel information S_B^{idx}, S_C^{idx} and S_C^{pxl}, and the sets of coefficients Ao, Bo, Ag, and Bg. Second, the image Y is segmented to obtain the same superpixels S and clusters C obtained in the hiding stage. With the recovering information S_C^{idx}, S_C^{idx} and S_C^{idx} we can generate the final segmentation $Segs$. For each segment in $Segs$ of luminance Y, the recovered channels Co' and Cg' were calculated using the coefficients in Ao, Bo, Ag, and Bg. Finally, the luminance Y and chrominance Co' and Cg' are transformed into the color space RGB to obtain the colorized image I', with the following equation:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}, \tag{2}$$

The proposed scheme is based on some previous algorithms. In the protection stage, we separate the color information from the luminance based on the linear approximation reported in [22]. Then, the color information is hidden in the luminance using the RDH-CE approach described in [20]. During the hiding process, the luminance contrast is altered to hinder the illegal recovery of the original color image. Finally, the parameters of the previous schemes are optimized using the GWO algorithm proposed in [21] to vary the protected image distortion. To protect the copyright of the image, the user can embed into the original luminance some visible-imperceptible watermarks [30] that contain the logo of the owner. The watermark was revealed using the RDH-CE hiding process. All previous protection stages are reversed by extracting the hidden color information, recovering the original luminance contrast, and obtaining a colorized version of the original image. Again, the watermark is visually imperceptible in the recovered colorized image.

Although our proposal is based on previous algorithms, they were modified to improve results. Fig. 6 shows the algorithms that were adapted or modified (marked with an asterisk *) with respect to the original versions. Firstly, the objective function of GWO method is defined such that the quality of the protected image is modified. Additionally, there are some differences between the proposed colorizing scheme shown in Fig. 1 and the method described in [22]. The color space selected is $YCoCg$ instead of the personalized YC_1C_2 defined in [22]. Then, the segmentation in [22] comprises a bilateral filter, followed by the SLIC superpixel scheme [27]. In contrast, our segmentation stage comprises a total variation [31] and Laplacian filter, followed by an LSC superpixel method [29]. In addition, the segmentation process is improved by including a clustering approach and a method that detects and corrects faulty colorized segments. Then, the color information was binarized using a precision-based scheme instead of the Huffman coding used in [22]. Finally, although RDH and watermarking algorithms are not modified individually, we solved some problems to make them coexist. Specifically, the watermark approach must not alter the capacity of the RDH algorithm so the color can fit in the luminance.

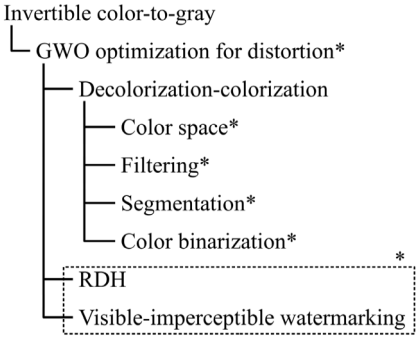


FIGURE 6. Algorithms used and improved (*).

B. SUPERPIXEL AND CLUSTERING SEGMENTATION

To further improve the segmentation adherence to the borders, we implement a preprocessing to the image before superpixel segmentation. Preprocessing consists of two filters. First, the total variation method [31] filters the luminance image by removing high frequencies with low degradation at the borders. This helps obtain smoother superpixel shapes in high-texture areas. Two parameters must be defined: the number of iterations Ti and the amount of denoising λ, which is commonly in the range $[0.01, 10]$ and is more aggressive with fewer values. Second, the local Laplacian filter method [32], [33] increases the local contrast and helps emphasize borders in low-contrast areas. The filter has three parameters: $\sigma \in [0, 1]$ controls the amplitude of edges, α typically in $[0.01, 1]$ adjusts the contrast, and $\beta \in [0, 1]$ controls the dynamic range. Both preprocessing algorithms were selected because of their good border preservation.

In the present proposal, we segmented the luminance channel of the image into superpixels using the LSC method [29]. The LSC is a clustering-based superpixel algorithm that generally exhibits good boundary adherence and pixel regularity. In LSC, the desired number Ns of superpixels can be selected, and the algorithm adjusts the boundary adherence with a defined parameter r in the common range $[0.05, 0.3]$.

In the hiding process, it is important to have fewer segments to reduce the quantity of the color information. Therefore, a clustering method is proposed to connect the superpixels and reduce the number of segments. To initialize the clustering, the first cluster is equal to the first superpixel area. The cluster is grown by including adjacent superpixels that are similar to the current cluster. In other words, if the difference between the means of the superpixel and cluster is less than the similarity threshold th, the superpixel is included in the cluster. The current cluster continued to grow until no more similar superpixels are found. The next cluster is the next superpixel that does not belong to a cluster and grows

according to the previous description. The iterative process is completed when all the superpixels belong to a cluster.

To facilitate adjacency detection, we propose generating a graph of superpixels represented by an adjacency matrix. Each element of the adjacency matrix a in position (i, j) is calculated as follows:

$$a(i,j) = \begin{cases} 1 \text{ , } if \text{ } superpixels \text{ } i \text{ } and \text{ } j \text{ } adjacents, \text{ } and \text{ } i \neq j \\ 0 \qquad\qquad otherwise \end{cases} \tag{3}$$

If we want to know which superpixels are adjacent to superpixel i, we find each position j in the adjacent matrix, where $a(i, j) = 1$. Algorithm 1 presents the pseudocode of the clustering process.

Algorithm 1 Superpixel Clustering

Inputs: Luminance Y, superpixels S and threshold th
Outputs: Clusters C
Create the adjacency matrix a of superpixels
For each superpixel
  If the superpixel is not assigned to any cluster
    Create a new cluster equal to the current superpixel
    continue := True
    While *continue* = *True*
      Use the adjacency matrix to identify the adjacent superpixels to the cluster
      For each adjacent superpixel
        Calculate the mean intensity of the adjacent superpixel
        Calculate the mean intensity of the current cluster
        Calculate the difference d between the mean intensities of the superpixel and current cluster.
        Add to the cluster area the superpixel area if complies with $|d| < th$
      end For
      If nonadjacent superpixel complied with $|d| < th$, then *continue* = *False*
    end While
  end If
end For

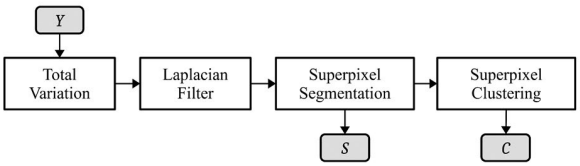


FIGURE 7. Luminance segmentation.

Fig. 7 shows the general process for image superpixels and clustering segmentations. It is important to note that preprocessing obtains segmentations C and S,and must not affect the original luminance Y where the color information is hidden.

C. FAULTY SEGMENTS DETECTION AND CORRECTION

The colorization process can be performed using superpixels or clusters. However, the recovered colorized image may contain artifacts or visual effects that do not correspond to the original image. The reason for this is explained by the images shown in Fig. 8. Figs. 8(a) and (b) show images of the COCO database [28] and a zoomed area, respectively. In addition, Fig. 8(c) shows the luminance of the zoomed area, where we can see how different colors may yield similar luminance values. Because segmentation is applied to luminance, the two colors may not be distinguished properly, and some segments may be inadequate for colorization.

FIGURE 8. (a) Color image. (b) Zoomed area. (c) The luminance of the zoomed area.

Figs. 9(a)–(c) show the superpixel, clustering, and final segmentation, respectively. The final segmentation is obtained using the proposed correction algorithm. Figs. 9(d)–(f) show the colorization of the three previous segmentations. Finally, Figs. 9(g)–(i) show the completely colorized images with the PSNR quality metric.

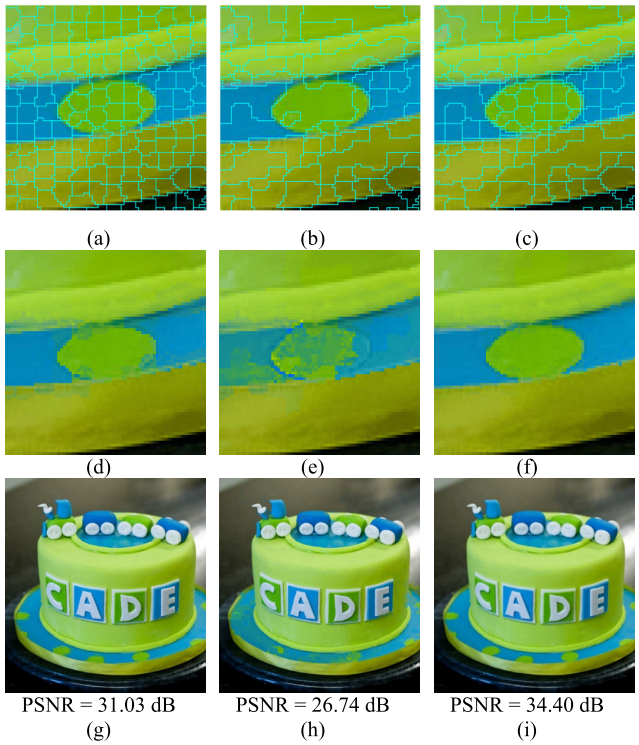


FIGURE 9. (a-c) Segmentation, (d-f) colorized region, and (g-i) colorized image by using superpixels, clusters, and improved segments, respectively.

To obtain the final segmentation, we proposed using both superpixels and clusters in a two-step process. First, the

luminance is colorized using the basic scheme shown in Fig. 4 with cluster segmentation. The areas of the superpixels that were not properly colorized are detected and included in cluster segmentation. Second, superpixels that cannot improve after colorization when included are detected and segmented into two parts. This is because these superpixels do not improve because they contain the image borders.

Fig. 10(a) shows the process used to obtain the final segmentation. The inputs of the process are color image I, superpixels S, and cluster C. The output includes the indexes S_B^{idx} of the faulty superpixels S_B to be included, the indexes S_C^{idx} of the faulty superpixels S_C to be segmented, and the binary segmentation S_C^{pxl} of the superpixels S_C to obtain the S_D segmentation.

We find the faulty superpixels using the following steps. First, we achieve the color approximation and colorizing schemes shown in Fig. 4 by using cluster C for colorizing segmentation. Then, the colors of all superpixels S in colorized image I' are compared with the colors of original image I. The comparison consists of calculating the mean square error (MSE) *MSEo* between the chrominance *Co* values in I and I' in the superpixel regions and the MSE *MSEg* between the chrominance *Cg* values in I and I'. A superpixel is considered faulty or non-well colorized if any of the errors *MSEo* and *MSEg* are greater than the normalized thresholds *tho* and *thg*, respectively, defined as follows:

$$\begin{aligned} tho &= (th_1 + th_2) \cdot (maxo - mino) + mino, \\ thg &= (th_1 + th_2) \cdot (maxg - ming) + ming, \end{aligned} \tag{4}$$

where *maxo* and *mino* are the maximum and minimum values *MSEo* for all superpixels, respectively. Similarly, *maxg* and *ming* correspond to values *MSEg*. Values th_1 and th_2 control the thoroughness of the evaluation. For the first superpixel evaluation, $th_1 \neq 0$ and $th_2 = 0$.

Subsequently, the areas of the detected faulty superpixels S_A are removed from the areas of cluster C, resulting in cluster C_A. The new coefficients approximation and the colorizing process are achieved with the new segmentation formed by the intersection $S_A \cup C_A$. The resulting colorized image I' is evaluated in the areas of each faulty superpixel in S_A. The superpixels in S_A are then separated into two sets: S_B and S_C. S_B contains superpixels that improve the quality according to the thresholds in (1), and S_C contains superpixels that do not improve. The thresholds in (1) must be incremented to make this second evaluation stricter; therefore, $th_1 \neq 0$ and $th_2 = 0$. Finally, superpixels in S_C are segmented into two parts to obtain the set of segments, S_D. Segments C_A, S_B, S_C and S_D are represented in cyan, yellow, red, and white, respectively, in the block diagram of Fig. 10(a) and in the color image of Fig. 10(b). The information S_B and S_C of the segments are necessary for the final segmentation recovery and must be hidden in the luminance using the RDH-CE process. To reduce the amount of information, we assign the number (index) of superpixels in S_A that belong to S_B and S_C, to the vectors S_B^{idx} and S_C^{idx}, respectively. Both vectors were

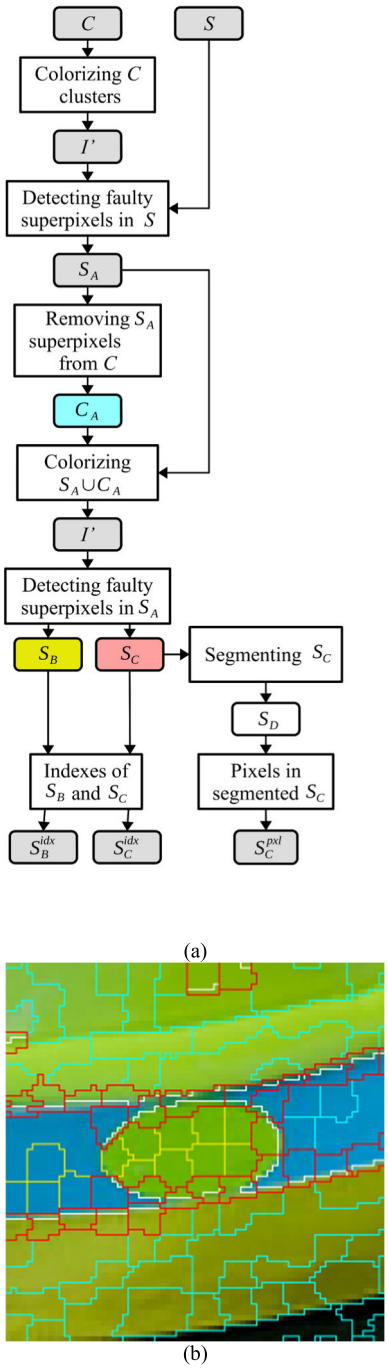


FIGURE 10. (a) The process to obtain the final segmentation in the color hiding stage. (b) Final segmentation with different colors for C_A (cyan), S_B (yellow), S_C (red) and S_D (white).

then binarized and concatenated to the information hidden in the luminance. In addition, a binary vector S_C^{pxl} contains pixels that belong to the first and second segments of the segmented superpixel S_D.

Fig. 11 shows the process for recovering the final segmentation *Segs*. Clusters C and superpixels S were obtained from luminance. The superpixels S_A, S_B and S_C are identified

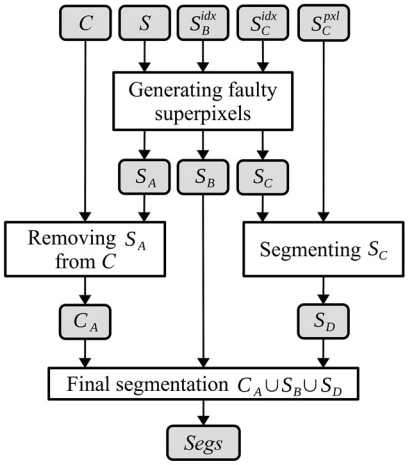


FIGURE 11. The process to obtain the final segmentation in the colorizing stage.

using the indexes S_B^{idx} and S_C^{idx}. Then, the superpixel S_A areas are removed from cluster C to obtain the segment C_A. The superpixels in S_C are segmented into two parts using the binary information in S_C^{pxl}. Finally, the final segmentation $Segs$ is defined as the union of the sets C_A, S_B and S_D.

Segmentation of the superpixels S_C was achieved using the k-means algorithm. In a superpixel, the vector $[co, cg]$ for each pixel contains two elements corresponding to its Co and Cg channel values. We then define two initial k-mean centroids: [40, 40] and [0, 0]. Finally, the k-means algorithm assigned each pixel vector to one of the k-means segments. We do not use threshold-based segmentation because we would require calculating Co and Cg thresholds separately and implementing post-processing to find one threshold. In addition, a superpixel may contain a few pixels, which may hinder the threshold calculation. Therefore, we selected the well-known k-means solution, which separates pixels using Co and Cg values together.

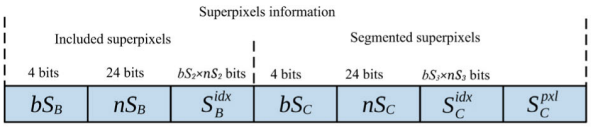


FIGURE 12. Binary string that contains the superpixel information.

The information to be hidden is organized as shown in Fig. 12. bS_B is the necessary number of bits to represent each superpixel index in S_B^{idx}, nS_B is the number of elements in S_B^{idx}, bS_C is the necessary number of bits to represent each superpixel in S_C^{idx}, and nS_C is the number of elements in S_C^{idx}. S_C^{pxl} is a binary vector whose elements are the segmentation S_D of the superpixel S_C. The number of bits in S_C^{pxl} belonging to each superpixel is given by the number of pixels in the superpixel.

The numbers of necessary bits are calculated as follows:

$$\begin{aligned} bS_B &= \lceil \log_2 (maxS_B) \rceil, \\ bS_c &= \lceil \log_2 (maxS_C) \rceil, \end{aligned} \tag{5}$$

where $maxS_B$ and $maxS_C$ are the maximum superpixel index in S_B^{idx} and S_C^{idx}, respectively. $\lceil \cdot \rceil$ is the nearest integer greater than or equal to the argument,

D. COLORIZING SCHEME

The color information of each segment is extracted and compressed as follows: Considering n pixels in a segment, the $n \times 1$ column vector $\boldsymbol{Y}$ contains all luminance values, and the vectors $\boldsymbol{C_o}$ and $\boldsymbol{C_g}$ contain the corresponding chrominance values. The objective is to find two scalar values, A and B that approximate the chrominance values from luminance $\boldsymbol{Y}$ as follows:

$$\boldsymbol{C_x}' = A + B\boldsymbol{Y}, \tag{6}$$

where $\boldsymbol{C_x}'$ can be any chrominance approximation vector $\boldsymbol{C_o}'$ or $\boldsymbol{C_g}'$.

The simple linear regression problem in (6) can be solved by minimizing the sum of the squares of the differences between the values in any original chrominance vector $\boldsymbol{C_x}$ and linear function $\boldsymbol{C_x}'$. The solution is commonly found by the ordinary least squares (OLS) method [34] using the following matrix operation:

$$\begin{bmatrix} A \\ B \end{bmatrix} = \left(\boldsymbol{Y_2^T Y_2}\right)^{-1} \boldsymbol{Y_2^T C_x}, \tag{7}$$

where $\boldsymbol{Y_2}$ is an $n \times 2$ column vector, the first column is a vector of ones, and the second column contains the luminance values $\boldsymbol{Y}$.

The A and B coefficients of each segment for Co channel are included in the sets Ao and Bo, respectively. Similarly, the coefficients for Cg channel are included in the sets Ag and Bg. The coefficient information is then binarized and hidden in the luminance. Therefore, we must binarize the coefficients such that the amount of data fits the luminance. However, we must maintain the precision of the coefficients sufficiently to obtain good color quality in the colorized image. Therefore, the precision step is a parameter that controls the size of color information.

For clarity, the example in the following description is referred to as coefficient A, note that the process can be applied to any coefficient B. The precision of a set of values A is defined by the step value P_A, which is the difference between a set of rounded values. The objective is to reassign each coefficient A to the nearest value in the rounding set. It is important to note that the rounding values were organized to include zero. That is, if $P_A = 0.2$ then the rounding values are in the set $\{\cdots -0.4, -0.2, 0, 0.2, 0.4, \cdots\}$. For example, if $A = 2.345$ and $P = 0.1$, the recovered value A' is rounded to $A' = 2.3$, and if $P_A = 0.02$, then $A' = 2.34$.

The steps to transform a coefficient A to a binary value are the following:

1. The value A is divided by P_A and the result is rounded to the nearest integer, as shown in (8). This step transforms value A into integer A_I.

$$A_I = round\left(\frac{A}{P_A}\right), \tag{8}$$

2. Obtain the minimum $minA_I$ among all A_I values and subtract it from each A_I value, as shown in (9). This operation shifts all integers A_I to remove negatives. Therefore, A_P values are in the range $\{0, 1, 2, \ldots, maxA_p\}$ with $maxA_P$ as the maximum of all A_P values.

$$A_P = A_I - minA_I \tag{9}$$

3. Thus, integer binarization is applied to each integer A_P using L number of bits defined as follows:

$$L = \lceil \log_2 (maxA_P) \rceil, \tag{10}$$

where $\lceil \cdot \rceil$ is the nearest integer greater than or equal to the argument.

Fig. 13 shows a numerical example of the binarization and recovery of coefficients Ao.

Stage (Eq.)	Example
Precision Step	$P_A = 0.2$
Coefficients	Ao = [1.3 -0.8 0.5]
Integers (8)	Ao_I = [7 -4 3], $minA_I$ = -4
Positives (9)	Ao_P = [11 0 7], $maxA_p$ = 11
Binary length (10)	$L = 4$
Binary representation	1011 0000 0111
Positives recovered	Ao_p' = [11 0 7]
Coefficients recovered (11)	Ao' = [1.4 -0.8 0.6]

FIGURE 13. Binary string with the color coefficients information.

The binarization process is applied to each element in the sets Ao and Ag, separately, with the step value P_A. Similarly, the elements of sets Bo and Bg are binarized using a different step value, P_B. All the information is concatenated, as shown in Fig. 14. The final binary string is a part of the information hidden in the luminance.

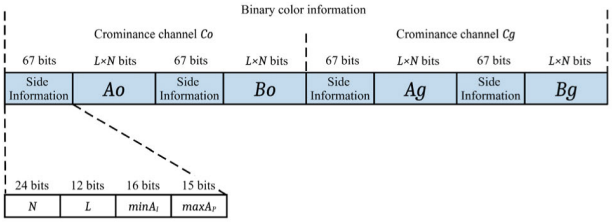


FIGURE 14. Binary string with the color coefficients information.

The information required to recover the approximated coefficients in Ao and Ag is the number of segments N, maximum length L, step P_A, minimum value $minA_I$ and maximum value $maxA_P$. The recovery process consists of extracting binary information from the luminance, calculating L from (10), and extracting N values of length L from the string. Each binary substring is then transformed into a recovered decimal integer A'_P. Finally, each recovered coefficient A' is calculated as follows:

$$A' = (A_P' + minA_I) \cdot P_A, \tag{11}$$

Similarly, the recovery of the coefficients in Bo and Bg is achieved by using their corresponding number of segments N, maximum length L, step P_B, minimum value $minB_I$ and maximum value $maxB_P$.

E. REVERSIBLE DATA HIDING WITH CONTRAST ENHANCEMENT (RDH-CE)

To the best of our knowledge, the first method to achieve RDH-CE has been reported in [35]. One of the last approach reported in [20] has a good preprocessing technique, which decreases the size of the side information for removal, reduces distortions to the image, and requires only one parameter H. The binary information is hidden in the image by shifting H times to the bins of the image histogram. The color information and segmentation data are hidden by using RDH-CE reported in [20] however, the information is hidden irregularly in the image according to the three hiding stages shown in Fig. 15(a). The first part of the information is hidden in areas with visible imperceptible watermarks when they exist. This is because we can increase the watermark visibility in the luminance by concentrating the hidden payload in these block areas. To increase the contrast effect, the RDH-CE [20] algorithm is applied many times until the capacity of the watermark block is reached. Second, the next part of the information is hidden in the entire image. Finally, the image is divided into $F \times F$ non-overlapping blocks, and the rest of the information is hidden in each block. In [36], it is shown that the capacity and quality of the marked image can be increased by applying RDH embedding in blocks instead of in a complete image. If the information is not completely hidden in all blocks, block division and hiding are iteratively repeated until all data are hidden. By contrast, Fig. 15(b) shows the inverse process. As expected, we must extract the hidden information backward from the image blocks, the entire image, and the watermark blocks, respectively.

Some information is required for reversibility, the watermark block positions, the number of iterations in watermark blocks, the total number of hiding processes in image blocks, and the length of the hidden binary information. This information can be hidden with colorization and segmentation information, or as part of an external key for recovery. The second option was used to obtain the results.

One characteristic of embedding in image blocks is that block artifacts may appear and increase at each embedding iteration, resulting in a marked image, as shown in Fig. 16(a). To diminish these artifacts, we propose an approach described in [37]. First, we select a small number H of histogram shifts to hide the information in the blocks. Second, we shift

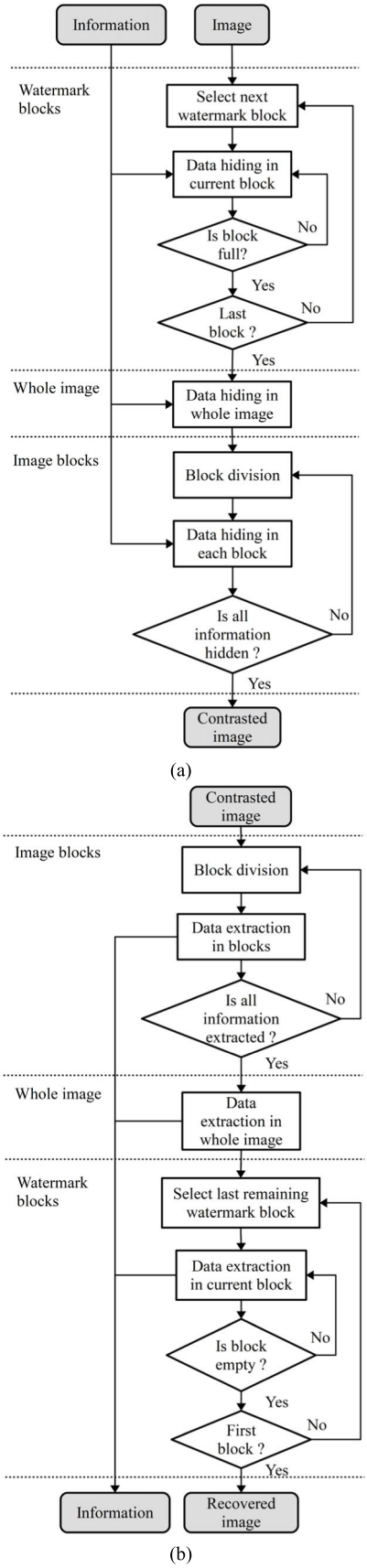


FIGURE 15. (a) Proposed data hiding to increase capacity and quality. (b) Proposed data extraction.

(a) (b)

FIGURE 16. Enhanced luminance with color hidden (a) without block shifting and (b) with block shifting.

the block positions at each embedding iteration to create an embedding in a different block. For example, for the first hiding iteration, we do not shift the positions of the blocks. Then, for the second iteration, the block positions are shifted vertically and horizontally to 1/2 their vertical and horizontal sizes, respectively. The third hiding iteration is achieved on the blocks shifted by 1/4 of their size, and so on. The shifting ratios for the blocks change at each iteration with the sequences 0, 1/2, 1/4, 3/4, 1/8, 3/8, 5/8, 7/8, 1/16, 3/16, 5/16, 7/16, ... This approach ensures that the block mesh is always different, and the borders of the blocks do not overlap and sharpen. Fig. 16(a) shows the result of RDH embedding when using $H = 62$, $F = 4$ and without the block-shifting method. Fig. 16(b) shows the result of RDH embedding with $H = 3$, $F = 4$, and with the block-shifting approach, we can see that block artifacts are considerably reduced. Using this approach, we can vary the quality of the contrasted luminance.

F. GREY WOLF OPTIMIZER (GWO) FOR PARAMETERS OPTIMIZATION

The GWO algorithm was first described in a previous study [21]. This optimization method is inspired by the hunting behavior of wolfs, which follows the members with the highest hierarchy when chasing prey. The optimization process tunes the parameter values of the algorithm at each iteration to obtain better results. The parameters of the optimizer are the number of agents, number of iterations, values to be tuned, range of these values, and the objective function. The number of agents selected for the GWO optimization is 35. Additionally, the number of iterations is initially defined as 100; however, the results show that fewer iterations may be acceptable according to the application. Table 1 lists the parameters of all the processes that achieve color hiding. The tunable parameters contain the optimization range, and the nontunable parameters are indicated. The proposed ranges were estimated experimentally over some images and applied to the full test dataset. To demonstrate the effectiveness of the values, Table 1 reports the ranges obtained in the experimental results of Section IV for all test images. The results were obtained for both optimization modes and 100 iterations.

TABLE 1. Tunable and non-tunable parameters.

Process	Symbol	Description	Range	Experimental range	
				Distorted	Balanced
Filtering	λ	Amount of denoising in total variation	[0, 3]	[0.57,2.82]	[0.42,2.93]
	Ti	Iterations (non-tunable)	100		
	σ	Edge amplitudes of local laplacian filter	[0, 1]	[0,0.50]	[0.01,0.94]
	α	Contrast of local laplacian filter	[0, 3]	[0.08,2.60]	[0.05,1.99]
	β	Dynamic range of local laplacian filter	[0, 3]	[0,2.78]	[0.02,2.40]
Superpixel	Ns	Ratio between number of superpixels and total of pixels	[0.004, 0.05]	[0.01,0.04]	[0.01,0.04]
	r	Superpixel adherence	[0.01, 1]	[0.02,0.99]	[0.05, 0.99]
Clustering	th	Similarity threshold to join segments	[1, 30]	[1,7]	[1,14]
Correction	th_1	Factor of color error for the first superpixel evaluation	[0.01, 0.5]	[0.01,0.14]	[0.01,0.5]
	th_2	Factor of color error for the second superpixel evaluation	[0, 0.5]	[0,0.11]	[0,0.37]
Colorizing	P_A	Precision step for coefficients A (non-tunable)	0.1 or 0.5		
	P_B	Precision step for coefficients B (non-tunable)	0.001 or 0.005		
RDH-CE	H	Number of histogram shiftings	[1, 30]	[16,30]	[1,30]
	F	Blocks per image side	[1, 10]	[5,10]	[1,5]

We can see that most of the resulting ranges are within the desired values.

Finally, the objective function f is defined according to one of two modes: distorted or balanced. First, we define a normalized quality metric Q_C based on the PSNR value $PSNR_C$ between the colorized image and original image, as shown in (12). Additionally, we define a normalized quality value Q_Y as a function of the SSIM quality $SSIM_Y$ between the contrasted luminance obtained by the proposed RDH-CE process and an equalized image obtained by conventional histogram equalization [38], as shown in (13). Thus, the objective function is defined as shown in (14) for the distorted mode, and (15) for the balanced mode. Equation (14) depends only on the value Q_C related to the colorized image, and (15) depends on the weighted sum of Q_C and the value Q_Y of the contrasted luminance.

$$Q_C = 1 - \frac{PSNR_C}{50}, \tag{12}$$

$$Q_Y = 1 - SSIM_Y, \tag{13}$$

$$f = Q_C, \tag{14}$$

$$f = \frac{Q_C + Q_Y}{2}, \tag{15}$$

G. VISIBLE-IMPERCEPTIBLE WATERMARK

The areas to be embedded in the visible imperceptible watermark were selected by the user. The watermark was embedded in the original luminance and then revealed by the contrast increment of the RDH-CE algorithm. RDH-CE may emphasize noise and image structures that can interfere with watermark revealing. Thus, we recommend that the user select dark or bright blocks in the planar regions. In addition, before watermark embedding, a total variation filter [31] was applied to the selected block. The recommended number of iterations and amount of denoising were $Ti = 100$ and $\lambda = 1$, respectively. Filtering a small region of the image may cause the average value to move with respect to the original region, resulting in a visible block effect. To match the block with the surroundings, a mapping method is applied after the total variation filter, considering the filtering yield values in the range [0,1]. The three methods were tested by multiplying the total variation output in the range [0,1] per 255, mapping the minimum and maximum values in the range [0,255], and a linear approximation, as shown in Fig. 17.

(a) (b) (c) (d)

FIGURE 17. Each image contains the grayscale image and a Sobel edge detector. (a) Original image. Image with filtered block (b) multiplied by 255, (c) mapped in the range [0, 255], and (d) mapped by a linear approximation.

As shown in Fig. 17(d), the linear approximation considerably reduced the edge effect of the selected block. Therefore, the pixels in the filtered block are recalculated as follows:

$$\boldsymbol{F} = a + b \times \boldsymbol{T}, \tag{16}$$

where $\boldsymbol{F}$ and $\boldsymbol{T}$ are blocks with filtered and total variation values, respectively. Scalars a and b are calculated using the OLS operation.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \left(\boldsymbol{T}^T \boldsymbol{T}\right)^{-1} \boldsymbol{T}^T \boldsymbol{B}, \tag{17}$$

where the first column of $\boldsymbol{F}$ is a column vector of ones and the second column contains the total variation values. $\boldsymbol{B}$ is a column vector containing the values of the original block.

Finally, a visible imperceptible watermark was embedded in the filtered blocks. To the best of our knowledge, the first method that reveals visible-imperceptible watermarks using RDH-CE was reported in [30]. However, we modified the embedding equation using composition (18), first proposed in [39]. B is the filtered block, W is the resized watermark, α is the embedding strength, and B_W is the watermarked block. The watermark strength α is defined in the range $[0, 255]$, which avoids overflows and maintains a balanced contribution from both images.

$$B_W = (1-\alpha)\,B + \alpha W, \tag{18}$$

We recommend calculating the strength α for bright and dark blocks as follows:

$$\alpha = \begin{cases} 0.018, & \textit{if mean block value} > 128 \\ 0.016, & \textit{otherwise} \end{cases}, \tag{19}$$

IV. RESULTS AND ANALYSIS

In this section, the experimental results are presented in six sections. First, we evaluate the optimization process and analyze how the images quality change with different iterations. Then, we describe the integration of the visible imperceptible watermark and study its effectiveness. Once the different security schemes are defined, we test them over different images. We show that the modifications over the linear approximation colorization in [22] overcomes its results. To show that our scheme reaches good quality for colorized images, we compare the proposed colorizing method with other state-of-the-art invertible colorization schemes. Finally, a color security test is accomplished by colorizing the protected images with deep colorization [40]. 29 images were used to obtain the reported results: the first 24 images belong to the Kodak Lossless True Color Image Suite [41]. Each image is either 768×512 or 512×768 in size. Additionally, we used five images from the USC-SIPI dataset [42] of size 512×512, shown in Fig. 18.

FIGURE 18. Images in USC-SIPI for testing.

A. OPTIMIZATION EFFECTIVENESS

In the optimization process, the user selects objective function (14) or (15) according to the security mode. GWO optimization adjusts the tunable parameters in Table 1 to reduce the function value and change the qualities of the colorized and contrasted images. Fig. 19 shows the numerical metrics for different iterations. Each point on the curve is the mean

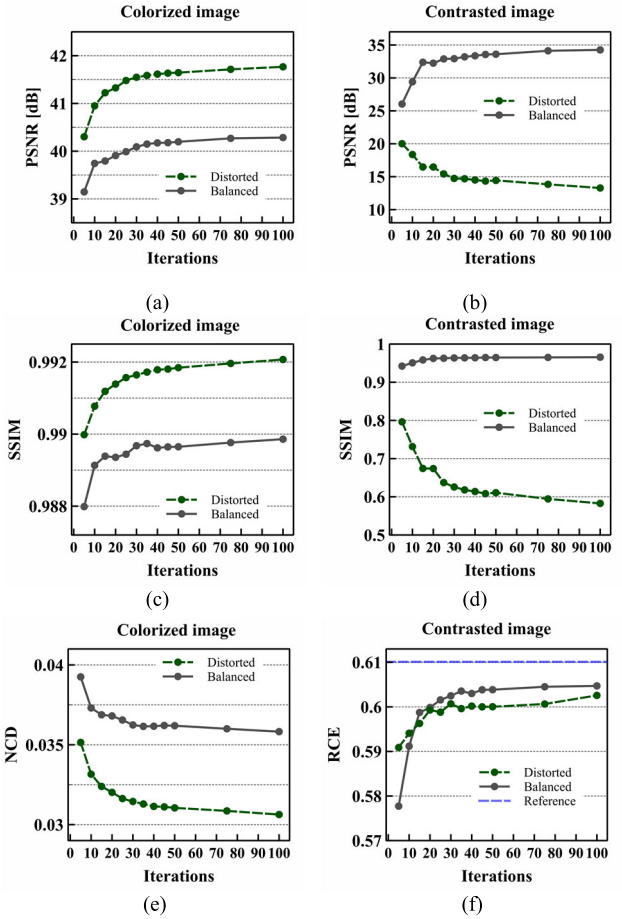


FIGURE 19. Mean quality improvement of the optimization process for the 29 images. (a) PSNR, (c) SSIM, and (e) NCD qualities for colorized images. (b) PSNR, (d) SSIM, and (f) RCE metrics, for contrasted images with respect to the reference luminance.

value obtained by applying the protection algorithm to all images. The green and gray curves correspond to distorted and balanced modes, respectively. Figs. 19(a)–(c) show the PSNR, SSIM, and NCD values, respectively, for the colorized images. As expected, the quality of the colorized image increases at each optimization step. The differences between the green and gray curves are approximately 1.5, 0.002, and 0.005 for the PSNR, SSIM, and NCD, respectively. This implies that the quality of the colorized images does not significantly decrease in the balanced mode. On the other hand, Figs. 19(d)-(e) show the PSNR and SSIM, respectively, of contrasted images with respect to the equalized image as reference. As expected, the image quality decreases in the distorted mode and increases in the balanced mode. To measure the contrast increment at different iterations, we show the RCE metric [43] in Fig. 19(f). RCE values above 0.5 reflect a contrast increment. The blue dashed line represents the ideal contrast value because it corresponds to the the RCE value of the reference luminance with respect to the original luminance.

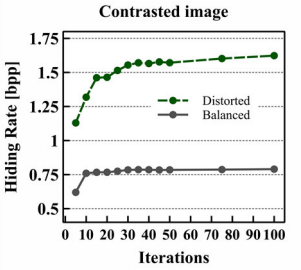


FIGURE 20. Images in USC-SIPI for testing.

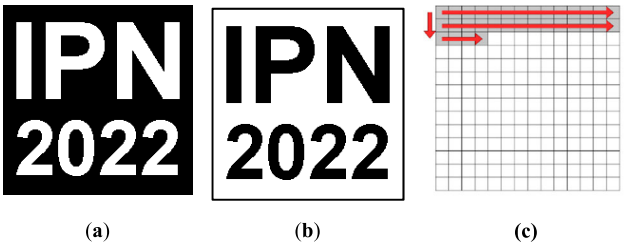


FIGURE 21. (a), (b) 200 × 200 watermark images. (c) Order of blocks to find the maximum area.

The hiding rate (HR) is shown in Fig. 20. The distorted mode requires a greater hiding capacity to increase the quality of the colorized image without considering the quality of the contrasted image. In contrast, the balanced mode maintains a lower hiding rate to maintain the contrast luminance quality.

B. VISIBLE-IMPERCEPTIBLE WATERMARK

After optimization, the user selects the watermark areas and the algorithm embeds visible imperceptible watermarks in the original luminance. The decolorization process then protects the color image using the optimized parameters. Consequently, the watermark is revealed in the resulting contrast luminance. The optimization process runs with no predefined visible-imperceptible watermark, so the final user can subsequently select the logo and areas. However, watermark areas reduce the luminance regions where the information is hidden. Therefore, the hiding capacity may decrease, and the color information obtained by the tuned parameters may no longer fit in the luminance. To address this problem, we change the precision of the colorizing coefficients from $P_A = 0.1$ and $P_B = 0.001$ with no watermark, to $P_A = 0.5$ and $P_B = 0.005$ with user watermarks. Increasing P_A and P_B values by 5 times, reduces the length of the color information and increases the maximum area to embed watermarks. In this section, we report the maximum approximated area that the user can select for watermarking using the recommended coefficient precision. Additionally, we demonstrate that this approach does not considerably decrease the quality of the colorized image.

To find the maximum area, we divided the luminance into non-overlapping blocks and then applied an iterative process. Each iteration consists of selecting subsequent number of blocks and embed a watermark in each. Then, the hiding process is achieved in all the image. Each iteration increases the number of blocks, and the process stops when the color data does not fit the luminance. The watermark in Fig. 21(a) and (b) are embedded in dark and white areas, respectively, according with the criterion in (17). Finally, the maximum area is the total area of the selected blocks in the last successful iteration. The area of each block is approximately 0.5% of the image area in a square shape, and the block selection is in raster order, as shown in Fig. 21(c).

Table 2 lists the percentages of the maximum reached areas for 20, 30, and 50 optimization iterations of the distorted mode. We compared the maximum areas for both the original and modified precisions, P_A, P_B. Table 2 lists the mean of the maximum values used to demonstrate the area increment. In addition, the minimum of the maximum reached values indicates that a sufficient area is guaranteed for all test images. Similarly, Table 3 lists the percentages of the maximum areas for the balanced mode.

TABLE 2. Maximum possible area for the visible-imperceptible watermark in the distorted scheme for the 29 images.

	Mean			Minimum		
Iterations	20	30	50	20	30	50
$P_A = 0.1, P_B = 0.001$	10.0%	7.3%	6.7%	<0.5%	<0.5%	<0.5%
$P_A = 0.5, P_B = 0.005$	15.8%	12.7%	11.3%	6.4%	7.4 %	2.5%

TABLE 3. Maximum possible area for the visible-imperceptible watermark in the balanced scheme for the 29 images.

	Mean			Minimum		
Iterations	20	30	50	20	30	50
$P_A = 0.1, P_B = 0.001$	68.5%	63.9%	56.1%	0.5%	<0.5%	<0.5%
$P_A = 0.5, P_B = 0.005$	89.7%	86.9%	89.4%	1.5%	7.4%	2.5%

To provide a visual reference of the area percentage, we show Figs. 22(a)–(c). The red borders in Figs. 22(a)–(c) represent 1.5%, 2.5%, and 7.5% of selected areas, respectively. In addition, the blocks contained visible imperceptible watermarks. Figs. 22(d)-(f) show the revealed watermarks in the contrasted luminances, with 30 iterations. To demonstrate that the areas are sufficient for watermark revealing, Figs. 22(g)–(i) show zoomed parts of the marked and revealed areas. The results in Fig. 22 demonstrate that 1.5% of the area is sufficient to hide and reveal the watermarks. To increase watermark revealing, we recommend using a logo with a black background, as shown in Fig. 21(a), in dark areas. In contrast, we recommend inverting the logo as shown in Fig. 21(b) to obtain better revealing in white areas.

C. EFFECTIVENESS OF THE PROPOSED SCHEME

Based on the quality values in Fig. 19, the selected number of iterations may depend on the quality requirements of the application. However, the following results were obtained with 30 iterations, where the quality values tended to stabilize.

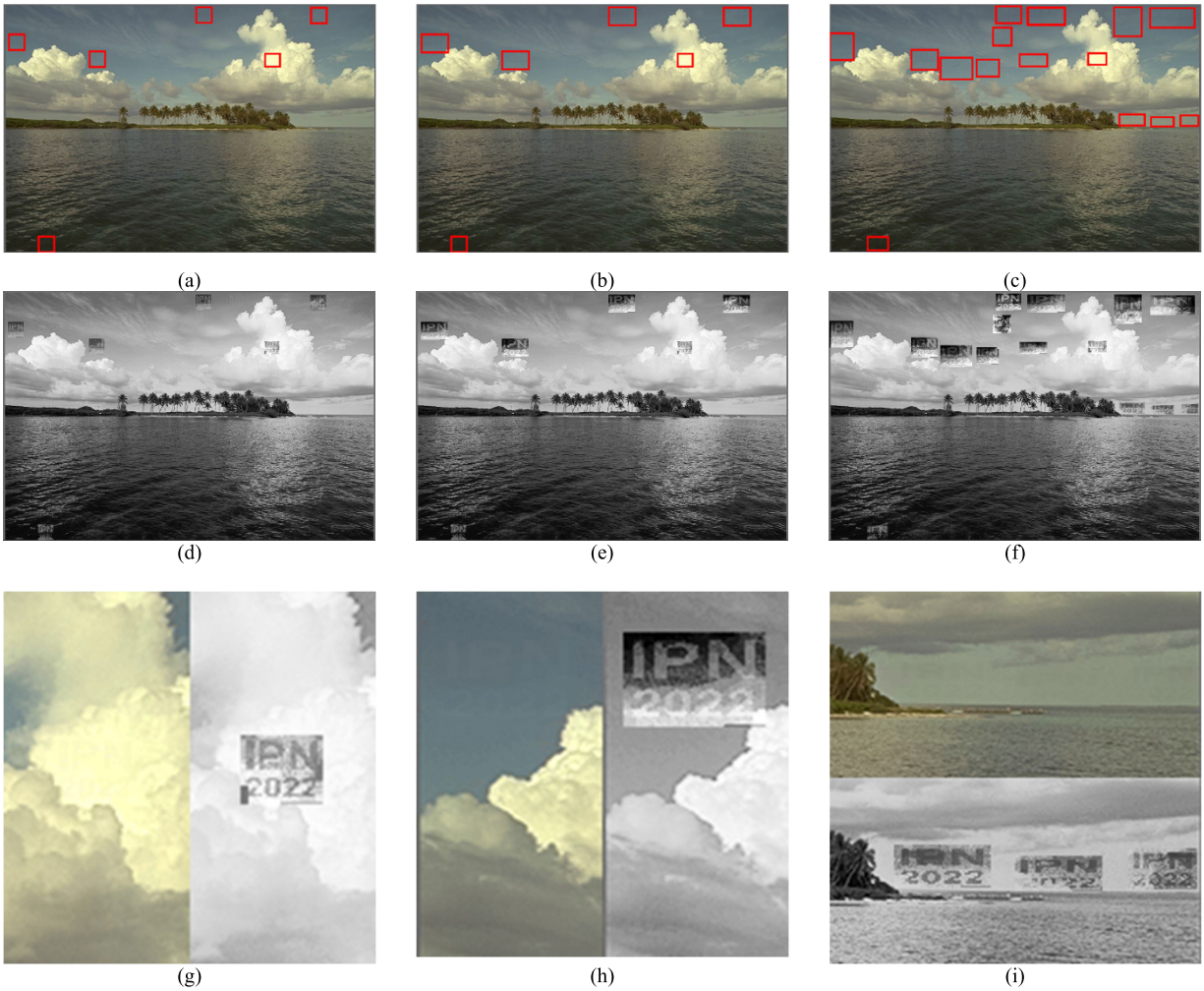


FIGURE 22. Colorized image with (a) 1.5%, (b) 2.5%, and (c) 7.5% of image areas, respectively, for the imperceptible watermark. (d), (e), (f) Contrasted images and (g), (h), (i) zoomed regions to demonstrate the imperceptibility and perceptibility of the watermark, by using 1.5%, 2.5%, and 7.5% of the area, respectively.

Tables 4 and 5 show a comparison of the qualities of all security options. Recall that the main objective is to obtain high-quality colorized images, low-quality contrasted images in distorted mode, and acceptable structural qualities for contrasted images in balanced modes. Table 4 shows the qualities of the colorized and contrasted images for the distorted and balanced modes when no visible imperceptible watermark is embedded. Similarly, Table 5 shows the qualities when visible-imperceptible watermarks are embedded in 5% of the image area. Firstly, we analyze the results between both modes: distorted and balanced. Thus, as we see in each table, the quality of the colorized image is slightly less in balanced mode than in distorted mode. On the other hand, the quality of the contrasted image considerably decreases in the distorted mode, as expected. Secondly, we analyze the effect of visible imperceptible watermarking on quality values. Thus, a comparison between Tables 4 and 5 shows that the quality of the colorized images is slightly reduced with the visible-imperceptible watermark.

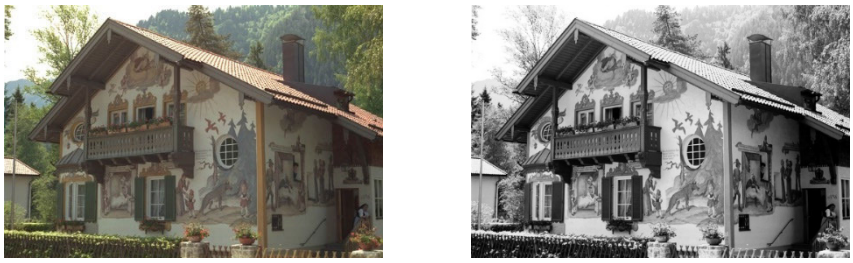
TABLE 4. Mean qualities of the proposed schemes with no visible-imperceptible watermark ($P_A = 0.1, P_B = 0.001$).

Mode	Colorized image			Contrasted image		
	PSNR [dB]	SSIM	NCD	PSNR [dB]	SSIM	RCE
Distorted	41.55	0.992	0.031	14.74	0.626	0.600
Balanced	40.09	0.990	0.036	32.92	0.964	0.602

TABLE 5. Mean qualities of the proposed schemes with 5% of the area for visible-imperceptible watermark ($P_A = 0.5, P_B = 0.005$).

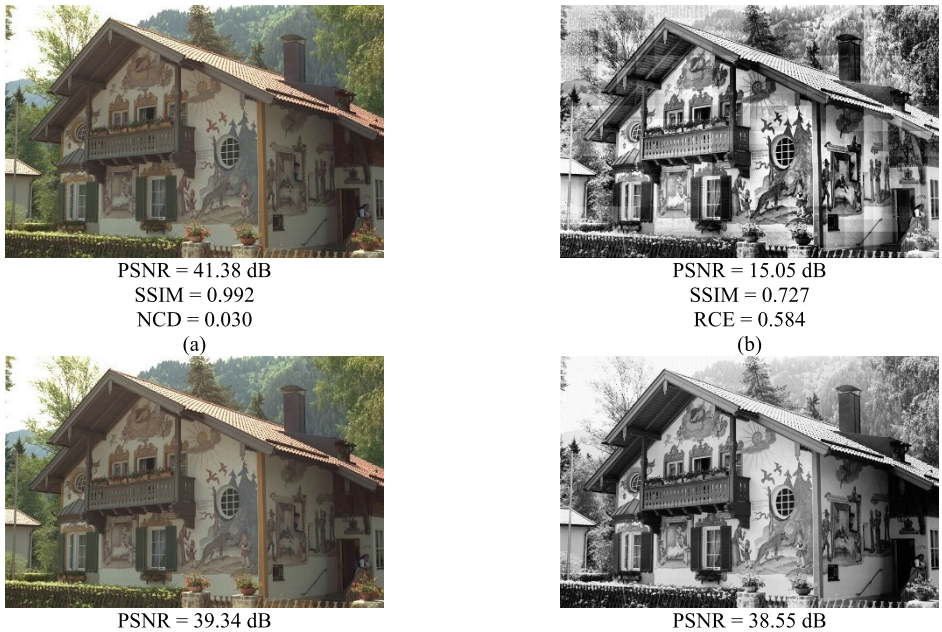
Mode	Colorized image			Contrasted image		
	PSNR [dB]	SSIM	NCD	PSNR [dB]	SSIM	RCE
Distorted	39.98	0.988	0.033	17.86	0.722	0.591
Balanced	38.87	0.987	0.037	22.60	0.916	0.577

The results that deserve deeper analysis are the qualities of contrasted image between non-watermark (Table 4) and watermark (Table 5) schemes. For example, we can see that



(a) (b)

FIGURE 23. (a) Original image used for demonstration and the corresponding (b) reference image.



PSNR = 41.38 dB
SSIM = 0.992
NCD = 0.030
(a)

PSNR = 15.05 dB
SSIM = 0.727
RCE = 0.584
(b)

PSNR = 39.34 dB
SSIM = 0.988
NCD = 0.037
(c)

PSNR = 38.55 dB
SSIM = 0.975
RCE = 0.583
(d)

FIGURE 24. Colorized (left) and contrasted (right) images without visible-imperceptible watermarks in (a), (b) distorted and (c), (d) balanced modes.

the PSNR of the contrasted image increases from 14.74 dB to 17.86 when using watermark in distorted mode. However, the quality decreases from 32.92 dB to 22.60 dB in balanced mode. To support explanation, we refer also to Fig. 2, where we can see a visual example in which the PSNR value in distorted mode increases, and in balanced mode decreases. Thus, two factors contribute to this effect. Firstly, the visible-imperceptible watermark tends to reduce contrasted image quality. Secondly, the reduction of the color information (given by P_A and P_B) when using watermark tend to increase the contrasted image quality, since less information is hidden in the luminance. However, in the distorted mode, the quality is already low, and the color information reduction can increase considerably the quality, even with the watermark. Therefore, the final effect is a quality increment. On the other hand, since the optimization in balanced mode yields the highest quality possible, the reduction of color information does not achieve to increase the quality considerably. Thus, the final effect is quality decrease caused by the watermark.

To show some of the visual results, we used the color image shown in Fig. 23. In addition, we demonstrate the equalized luminance used as a reference in the optimization.

Figs. 24 shows the visual and quantitative results without a visible imperceptible watermark for the distorted and balanced modes. Additionally, Figs. 25 shows the results with visible imperceptible watermarks for the distorted and balanced modes. The watermark areas are zoomed in to demonstrate the effectiveness of the imperceptibility. As we can see, the numerical results for each example, coincide with the mean values of Tables 4 and 5.

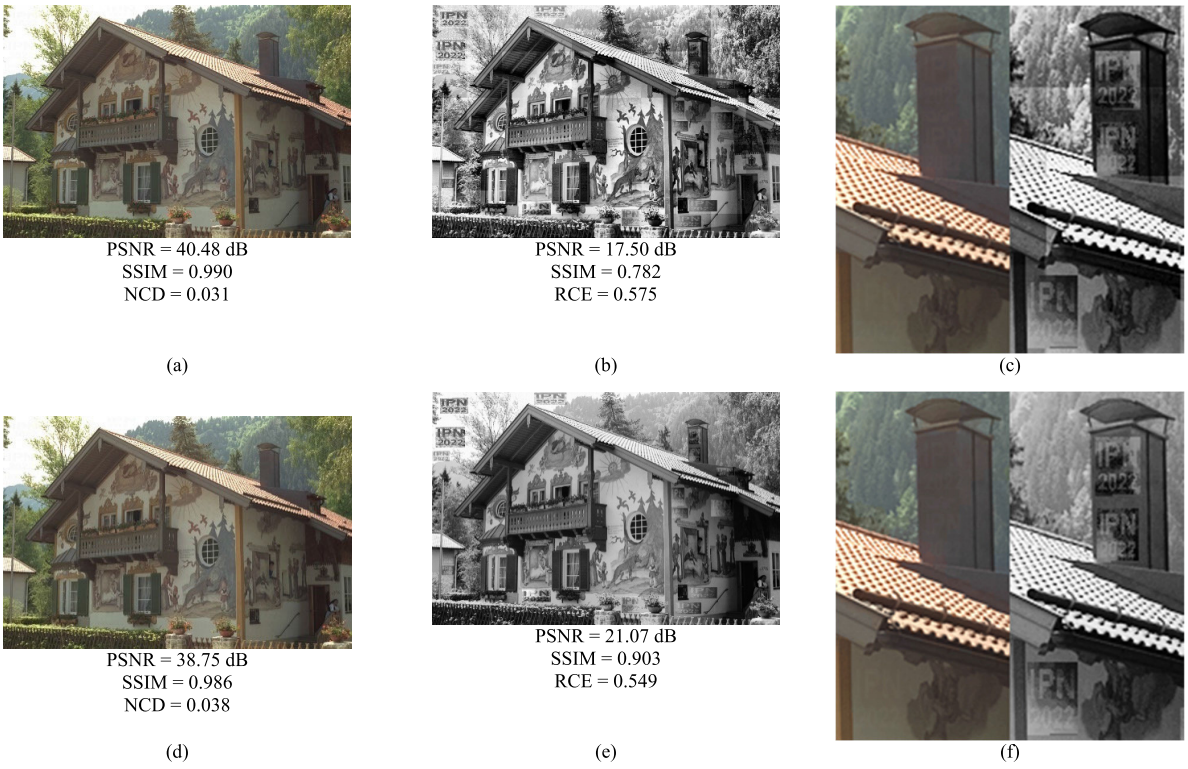


FIGURE 25. Colorized (left), contrasted (center) and zoomed regions (right) images with visible-imperceptible watermarks in (a), (b), (c) distorted and (d), (e), (f) balanced modes. The watermarks size is 5.5% of the image area.

FIGURE 26. (a), (d) Original image; (b), (e) Colorized image obtained with [22] and (c), (f) colorized image obtained with our proposal in distorted mode.

D. COMPARISON WITH LINEAR APPROXIMATION SCHEME

The proposed colorization algorithm is a modified version of the scheme in [22]. Therefore, we demonstrate that the modification of the original colorization scheme improves most of the qualitative and visual results. To achieve a fair comparison, we decided to pair the hiding rate (HR) of the previous scheme with the HR of the results in Fig. 19 for

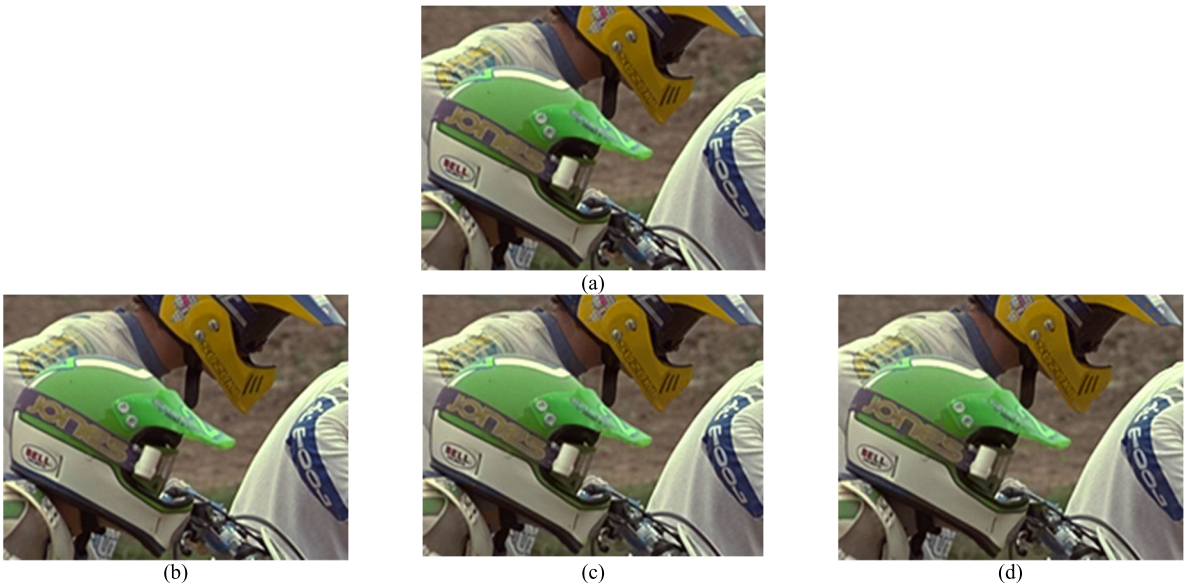


FIGURE 27. (a) Zoomed part of image in Kodak dataset. Colorized version obtained by (b) [19], (c) proposal in distorted mode and (d) proposal in balanced mode.

30 iterations. Therefore, we modified the number of superpixels [22] from a small value until the desired HR was achieved. Table 6 presents the results of this comparison.

TABLE 6. Mean qualities of the proposed schemes with no visible-imperceptible watermark ($P_A = 0.1, P_B = 0.001$).

Mode	HR [22] [bpp]	HR [bpp]	PSNR [22] [dB]	PSNR [dB]	SSIM [22]	SSIM	NCD [22]	NCD
Distorted	1.56	1.55	36.44	**41.548**	0.959	**0.992**	0.057	**0.031**
Balanced	0.79	0.78	37.90	**40.091**	0.982	**0.990**	0.045	**0.036**

To demonstrate the effectiveness of the correction in the segmentation, Figs. 26 shows an example of the colorization for both schemes. The images include a comparison between the colorized images of the previous scheme and those of our scheme. We can see that our proposed method has better color adherence to borders.

E. COMPARISON WITH OTHER INVERTIBLE COLOR-TO-GRAY APPROACHES

We propose to modify the structure of the gray-scale image to increase security. However, the main objective of previous proposals is to maintain the structural quality of the gray-scale version of the image. Therefore, we consider difficult even meaningless to compare the quality of the protected image. On the other hand, it may be valuable comparing the quality of the colorized image independently of the nature of the protected image. Table 7 shows the reported results in [19] for 11 proposals and our scheme. Table 7 shows the PSNR and FSIM [44] qualities for the colorized images, sorted in ascending order for PSNR.

TABLE 7. Comparison of mean qualities of the colorized images for Kodak dataset.

Method	PSNR [db]	FSIM
[1]	27.58	0.9914
[3]	33.13	0.9954
[4]	29.53	0.9981
[6]	33.07	0.9872
[10]	37.79	0.9953
[11]	32.71	0.9951
[12]	38.37	0.9985
[13]	36.44	0.9965
[15]	37.40	0.9958
Balanced	40.99	0.9988
[17]	42.20	0.9994
Distorted	42.52	0.9991
[19]	45.66	0.9994

Even with the results of the methods in [17] and [19], our scheme still shows good colorized image quality. Additionally, it offers additional security achieved by the gray-scale image distortion and the visible-imperceptible watermark. Figs. 27-29 show a comparison with the results of the scheme [19] in image parts with some difficult color and shapes structures. We can see that the visual differences may not be relevant since they are almost noticeable.

F. SECURITY TEST FOR COLORZIATION

In recent years, Deep Learning (DL) has provided diverse solutions in the field of image colorization. Popular DL approaches train a CNN by learning from a large dataset of color images; therefore, the net is capable of colorizing gray-scale images. Some of these schemes have two

FIGURE 28. (a) Zoomed part of image in Kodak dataset. Colorized version obtained by (b) [19], (c) proposal in distorted mode and (d) proposal in balanced mode.

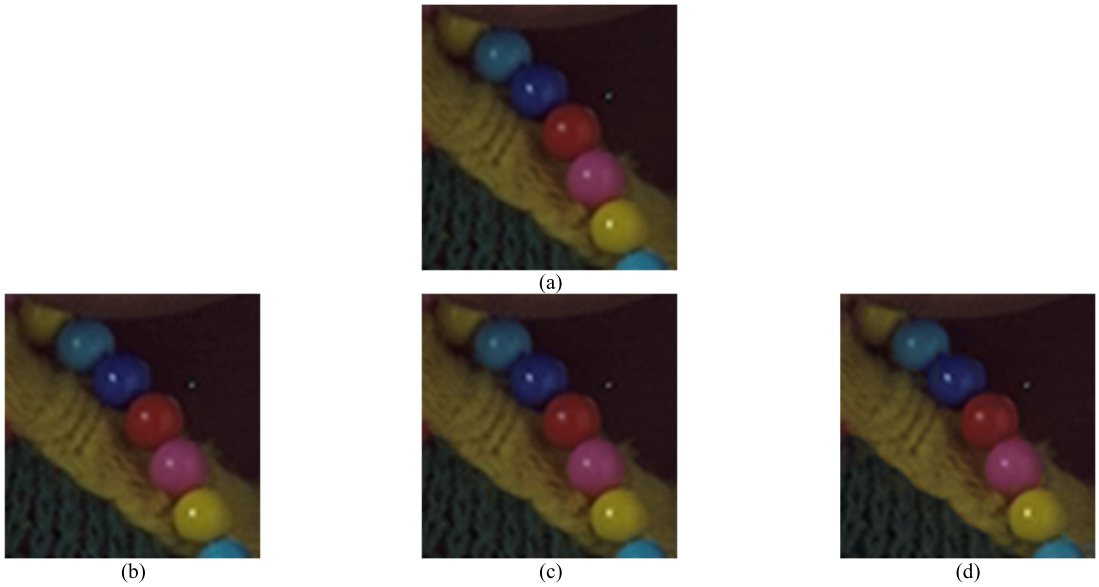


FIGURE 29. (a) Zoomed part of image in Kodak dataset. Colorized version obtained by (b) [19], (c) proposal in distorted mode and (d) proposal in balanced mode.

colorizing modes. First, DL algorithms can colorize images with net knowledge, without any color information of the original scene. Second, DL algorithms can use color references provided by the user to obtain adequate colorization. Therefore, CNN approaches may be a potential threat to illegally colorizing gray-scale protected images. Furthermore, if color information is illegally obtained, the CNN can use these data to obtain better colorization. Therefore, in this section, we demonstrate the difficulty of recovering the color of a protected image using DL. Thus, we demonstrate that we obtain better colorized images than the DL approach, even with color information.

The DL approach algorithm reported in [40] and [45] was used for comparison. Because of the effectiveness of the DL scheme, it has been included in Adobe Photoshop Elements® 2020 and the GIMP-ML suite [46] for the beta version of GIMP 2.99.6 [47]. Colorization can be applied with or without color reference. Fig. 30 shows two test images and a color reference at different points. Each color point is placed at the center of each segment generated by our

FIGURE 30. (a), (b) Original images and (c), (d) their corresponding color samples.

FIGURE 31. Colorized images obtained by (a), (b) deep colorization [40], [45] with no reference; (c), (d) deep colorization [40] with reference; and (e), (f) our proposal.

proposed algorithm, and its value is the mean of the color of the segment.

Figs. 31(a) and (d) show the colorized images from Figs. 30(a) and (c), respectively, using DL colorization with no reference. As expected, we could not recover most of the original colors because the color palette was selected according to the trained network. Then, Figs. 31(b) and (e) show the colorized images obtained using the referenced colors in Figs. 31(b) and (d), respectively. The colorized images do not appear to be saturated and have low color adherence to the borders. Finally, Figs. 31(c) and (f) show colorization using the proposed scheme.

TABLE 8. Mean qualities of colorized images obtained by proposal in [40] and our proposal in distorted mode.

	PSNR [dB]			SSIM			NCD		
	Non-ref [40]	Ref [40]	Proposal	Non-ref [40]	Ref [40]	Proposal	Non-ref [40]	Ref [40]	Proposal
Distorted	22.22	25.71	41.55	0.647	0.853	0.992	0.334	0.207	0.031
Balanced	22.22	26.10	40.09	0.647	0.866	0.990	0.334	0.196	0.036

Table 8 shows the mean qualities obtained by the non-referenced and referenced DL approaches compared with the present proposal.

V. DISCUSSION

The present proposal toggles between a distorted gray-scale image and a colorized image to protect the image structural content. Additionally, we embedded a visible-imperceptible watermark in the colorized image, which reveals the owner logo in the protected image. Therefore, we tested the distortion of the protected image and the quality of the colorized image with the different scenarios. A summary can be seen in Tables 4 and 5 for the recommended 30 iterations in the optimization. Firstly, the mean RCE contrast values are greater than 0.5, proving the contrast increment. Secondly, the mean PSNR values between 14.74 dB and 32.92 dB show the quantitative distortion of the protected image, which varies according to both distorted or balanced modes. Additionally, SSIM values between 0.626 and 0.964 indicate that the general structure of luminance remains. Thirdly, the quality of the colorized images is proved with the PSNR, SSIM and NCD mean values in the ranges [38.87 dB,41.55 dB], [0.987,0.990], and [0.031, 0.037], respectively. The narrow ranges show that the quality of the colorized images tends to remain regardless of the level of distortion in the gray-scale image or the usage of the visible-imperceptible watermark. Finally, the integration of visible-imperceptible watermarks became a problem because they hamper color hiding. Therefore, with the strategy presented in Section IV-B, we achieved to embed visible-imperceptible watermarks in more than 7.4% of the total area, which is sufficient to reveal the ownership of the content.

On the other hand, the colorization scheme was based on a previous study [22] focused on image compression. To achieve a fair comparison, we modified the number of segments in the previous scheme to match the HR values and compare the quality of the colorized images. This is because the original colorization approach was designed for compression and different applications have different assessments. For example, restoration methods focus on improving the quality of the colorized images. Compression approaches include an assessment of the hiding rate, which should be the minimum possible. Additionally, color-to-gray schemes involve protected image quality, which reflects content visibility and level of protection. Therefore, in future work, a unified assessment of isolated colorization algorithms can be proposed for fair comparison. A possible assessment criterion is to measure the quality of the colorized image according to the different amounts of color information. Despite the differences between the approaches, we compared our colorization improvement with that of the scheme in [22]. Table 6 shows that our strategy improved the quality of the colorized images. For example, with an HR value of 1.55 bpp in the distorted mode, we increased the PSNR, SSIM, and NCD qualities to 5 dB, 0.033, and 0.026, respectively. Furthermore, Fig. 26 shows a visual example of colorization improvement.

Additionally, the algorithm was compared with previous invertible color-to-gray approaches. Table 7 show that our approach reaches mean PSNR values above 40.99 dB, which is a common acceptable value. Just both proposals [17] and [19] obtain better colorized qualities in some cases, however, our proposal achieves the additional security obtained by distorted protected image and the copyright protection of the visible-imperceptible watermark. Additionally, the differences between quality values above 40 dB tend to yield negligible visual differences, as proved in Figs. 27 to 29 which compares our colorized images and the obtained in [19].

Finally, the results of section III.F show how illegal color recovery based on DL is not completely successful. Fig. 31 and Table 6 show that colorized images obtained by DL do not obtain the same quality as legal recovery, even when the original color information is available.

VI. CONCLUSION AND FUTURE WORK

The present proposal addresses the need to protect the color and structural content of an image in storage, transmission, or limited access environments achieved by the protected image distortion. Additionally, the ownership of the colorized image is protected by the visible-imperceptible watermark. The results show that the qualities of the protected and colorized images are competitive.

The modular design of our algorithm facilitates the improvement of general results by modifying isolated stages. Therefore, we identify that different solutions for segmentation, colorization, RDH-CE, and optimization may be explored in future work to improve the results. First, since the quality of the colorized image is closely related to segmentation, future work can involve the improvement of superpixels and clustering algorithms. Second, colorization based on linear approximation was selected as the first approach owing to its simplicity. However, we encourage future work to explore other colorization schemes to increase the color

quality. Additionally, we suggest exploring other types of distortions using different RDH or even robust watermarking techniques according to different possible applications. This will help hinder the illegal reconstruction of protected images and provide users with more security options. Finally, although the GWO method delivers successful results, some optimization algorithms may work more efficiently in the present application. Therefore, future work should explore and compare different optimization solutions to obtain better results. Finally, Table 1 shows a comparison between the desired range of tunable parameters and the range obtained in the experiments. Most of the desired ranges are adequate because the experimental values fall within these ranges. However, to improve efficiency, the limits of the ranges may be adjusted with more experiments over larger image datasets.

REFERENCES

[1] R. L. de Queiroz and K. M. Braun, "Color to gray and back: Color embedding into textured gray images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1464–1470, Jun. 2006, doi: 10.1109/TIP.2006.871181.

[2] R. L. de Queiroz, "Reversible color-to-gray mapping using subband domain texturization," *Pattern Recognit. Lett.*, vol. 31, no. 4, pp. 269–276, Mar. 2010, doi: 10.1016/j.patrec.2008.11.010.

[3] K.-W. Ko, D.-C. Kim, W.-J. Kyung, and Y.-H. Ha, "Color embedding and recovery using wavelet packet transform with pseudorandomized saturation code," *J. Imag. Sci. Technol.*, vol. 55, no. 3, May 2011, Art. no. 30501, doi: 10.2352/J.ImagingSci.Technol.2011.55.3.030501.

[4] T. Horiuchi, F. Nohara, and S. Tominaga, "Accurate reversible color-to-gray mapping algorithm without distortion conditions," *Pattern Recognit. Lett.*, vol. 31, no. 15, pp. 2405–2414, Nov. 2010, doi: 10.1016/j.patrec.2010.07.014.

[5] K.-W. Ko, O.-S. Kwon, C.-H. Son, and Y.-H. Ha, "Color embedding and recovery based on wavelet packet transform," *J. Imag. Sci. Technol.*, vol. 52, no. 1, Jan. 2008, Art. no. 10501.

[6] G. Tanaka, N. Suetake, and E. Uchino, "Invertible color-to-monochrome transformation based on clustering with lightness constraint," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2010, pp. 2151–2154, doi: 10.1109/ICSMC.2010.5641671.

[7] M. Chaumont and W. Puech, "A grey-level image embedding its color palette," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2007, pp. I-389–I-392, doi: 10.1109/ICIP.2007.4378973.

[8] M. Chaumont and W. Puech, "A fast and efficient method to protect color images," *Proc. SPIE*, vol. 6508, Jan. 2007, Art. no. 65081T, doi: 10.1117/12.702741.

[9] M. Chaumont and W. Puech, "Protecting the color information by hiding it," in *Recent Advances in Signal Processing*. Rijeka, Croatia: InTech, 2009, doi: 10.5772/7453.

[10] M. Chaumont, W. Puech, and C. Lahanier, "Securing color information of an image by concealing the color palette," *J. Syst. Softw.*, vol. 86, no. 3, pp. 809–825, Mar. 2013, doi: 10.1016/j.jss.2012.11.042.

[11] Z.-X. Xu and Y.-H. Chan, "Improving reversible color-to-grayscale conversion with halftoning," *Signal Process., Image Commun.*, vol. 52, pp. 111–123, Mar. 2017, doi: 10.1016/j.image.2016.12.005.

[12] Y. Chan, Z. Xu, and D. P. Lun, "A framework of reversible color-to-grayscale conversion with watermarking feature," *IEEE Trans. Image Process.*, vol. 29, pp. 859–870, 2020, doi: 10.1109/TIP.2019.2936097.

[13] Q. Liang, R. Hu, and S. Xiang, "Invertible color-to-grayscale conversion by using clustering and reversible watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2021, pp. 1–6, doi: 10.1109/ICME51207.2021.9428464.

[14] M. Chaumont and W. Puech, "A color image hidden in a grey-level image," in *Proc. CGIV 3rd Eur. Conf. Colour Graph., Imag., Vis.*, Jun. 2006, pp. 229–231.

[15] M. Xia, X. Liu, and T.-T. Wong, "Invertible grayscale," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–10, Dec. 2018, doi: 10.1145/3272127.3275080.

[16] T. Ye, Y. Du, J. Deng, and S. He, "Invertible grayscale via dual features ensemble," *IEEE Access*, vol. 8, pp. 89670–89679, 2020, doi: 10.1109/ACCESS.2020.2994148.

[17] R. Zhao, T. Liu, J. Xiao, D. P. K. Lun, and K. Lam, "Invertible image decolorization," *IEEE Trans. Image Process.*, vol. 30, pp. 6081–6095, 2021, doi: 10.1109/TIP.2021.3091902.

[18] K. Liu, D. Chen, J. Liao, W. Zhang, H. Zhou, J. Zhang, W. Zhou, and N. Yu, "JPEG robust invertible grayscale," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 12, pp. 4403–4417, Dec. 2022, doi: 10.1109/TVCG.2021.3088531.

[19] Q. Liang and S. Xiang, "Invertible color-to-grayscale conversion using lossy compression and high-capacity data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7373–7385, Nov. 2022, doi: 10.1109/TCSVT.2022.3184949.

[20] H.-T. Wu, S. Tang, J. Huang, and Y.-Q. Shi, "A novel reversible data hiding method with image contrast enhancement," *Signal Process., Image Commun.*, vol. 62, pp. 64–73, Mar. 2018, doi: 10.1016/j.image.2017.12.006.

[21] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[22] T. Fujisawa and M. Ikehara, "Color image coding based on linear combination of adaptive colorspaces," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 1522–1526, doi: 10.1109/ICASSP.2017.7952411.

[23] M. H. Noaman, H. Khaled, and H. M. Faheem, "Image colorization: A survey of methodolgies and techniques," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.*, 2022, pp. 115–130, doi: 10.1007/978-3-030-89701-7_11.

[24] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, Aug. 2004, doi: 10.1145/1015706.1015780.

[25] S. Lee, S. Park, P. Oh, and M. G. Kang, "Colorization-based compression using optimization," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2627–2636, Jul. 2013, doi: 10.1109/TIP.2013.2253486.

[26] K. Uruma, K. Konishi, T. Takahashi, and T. Furukawa, "Colorization-based image coding using graph Fourier transform," *Signal Process., Image Commun.*, vol. 74, pp. 266–279, May 2019, doi: 10.1016/j.image.2018.12.011.

[27] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012, doi: 10.1109/TPAMI.2012.120.

[28] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 8693. Cham, Switzerland: Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[29] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3317–3330, Jul. 2017, doi: 10.1109/TIP.2017.2651389.

[30] J. Sarabia-Lopez, D. Nuñez-Ramirez, D. Mata-Mendoza, E. Fragoso-Navarro, M. Cedillo-Hernandez, and M. Nakano-Miyatake, "Visible-imperceptible image watermarking based on reversible data hiding with contrast enhancement," in *Proc. Int. Conf. Mechatronics, Electron. Automot. Eng. (ICMEAE)*, Nov. 2020, pp. 29–34, doi: 10.1109/ICMEAE51770.2020.00013.

[31] W. Yin, D. Goldfarb, and S. Osher, "Image cartoon-texture decomposition and feature selection using the total variation regularized L1 functional," in *Variational, Geometric, and Level Set Methods in Computer Vision* (Lecture Notes in Computer Science), vol. 3752. Berlín, Germany: Springer, 2005, pp. 73–84, doi: 10.1007/11567646_7.

[32] S. Paris, S. W. Hasinoff, and J. Kautz, "Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–12, Jul. 2011, doi: 10.1145/1964921.1964963.

[33] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local Laplacian filters: Theory and applications," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–14, Sep. 2014, doi: 10.1145/2629645.

[34] I. Pardoe, , *Applied Regression Modeling*. Hoboken, NJ, USA: Wiley, 2012, doi: 10.1002/9781118345054.

[35] H. Wu, J. Dugelay, and Y. Shi, "Reversible image data hiding with contrast enhancement," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 81–85, Jan. 2015, doi: 10.1109/LSP.2014.2346989.

[36] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification," *IEICE Electron. Exp.*, vol. 4, no. 7, pp. 205–210, 2007, doi: 10.1587/elex.4.205.

[37] E. Fragoso-Navarro, M. Cedillo-Hernandez, F. Garcia-Ugalde, and R. Morelos-Zaragoza, "Reversible data hiding with a new local contrast enhancement approach," *Mathematics*, vol. 10, no. 5, p. 841, Mar. 2022, doi: 10.3390/math10050841.

[38] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. London, U.K.: Pearson, 2008.
[39] T. Porter and T. Duff, "Compositing digital images," *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 253–259, Jul. 1984, doi: 10.1145/964965.808606.
[40] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017, doi: 10.1145/3072959.3073703.
[41] R. Franzen. *Kodak Lossless True Color Image Suite*. Accessed: Aug. 11, 2021. [Online]. Available: http://www.r0k.us/graphics/kodak/
[42] *The USC-SIPI Image Database*. Accessed: Aug. 11, 2021. [Online]. Available: http://sipi.usc.edu/database/
[43] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3350–3364, Dec. 2011, doi: 10.1109/TIP.2011.2147325.
[44] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011, doi: 10.1109/TIP.2011.2109730.
[45] J.-Y. Zhu. *Interactive Deep Colorization*. Accessed: Oct. 1, 2022. [Online]. Available: https://github.com/junyanz/interactive-deep-colorization
[46] K. Soman, "GIMP-ML: Python plugins for using computer vision models in GIMP," *arXiv:2004.13060v3*, doi: 10.48550/arXiv.2004.13060.
[47] K. Soman. *GIMP-ML*. Accessed: Oct. 1, 2022. [Online]. Available: https://github.com/kritiksoman/GIMP-ML/tree/GIMP3-ML

EDUARDO FRAGOSO-NAVARRO (Member, IEEE) was born in Ciudad de Mexico, Mexico. He received the B.S. degree in mechatronics engineering, the M.S. degree in microelectronics engineering, and the first Ph.D. degree in communications and electronics from Instituto Politécnico Nacional (IPN), Mexico, in 2011, 2017, and 2021, respectively. He is currently pursuing the second Ph.D. degree with Facultad de Ingenieria, Universidad Nacional Autónoma de México (UNAM). His research interests include image processing, watermarking, and related fields.

FRANCISCO GARCIA-UGALDE was born in Mexico. He received the bachelor's degree in electronics and electrical system engineering from Universidad Nacional Autónoma de México (UNAM), in 1977, the Diplôme d'Ingénieur degree from SUPELEC, France, in 1980, and the Ph.D. degree in information processing from Université de Rennes I, France, in 1982. Since 1983, he has been a full-time Professor with UNAM. His research interests include video coding, image analysis, watermarking, theory and applications of error control coding, turbo coding, applications of cryptography, parallel processing, and data bases.

MANUEL CEDILLO-HERNANDEZ was born in Mexico. He received the B.S. degree in computer engineering, the M.S. degree in microelectronics engineering, and the Ph.D. degree in communications and electronic from Instituto Politécnico Nacional (IPN), in 2003, 2006, and 2011, respectively. He has six years of professional experience with government positions related to IT. From September 2011 to December 2015, he was with Facultad de Ingenieria, Universidad Nacional Autónoma de México (UNAM), where he was a Professor. Currently, he is a full-time Researcher with IPN. His research interests include image and video processing, watermarking, software development, and related fields.