www.ijemr.net ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Volume-7, Issue-3, May-June 2017

International Journal of Engineering and Management Research

Page Number: 164-171

Reversible Image Data Hiding with Contrast Enhancement

Aiswaria K R[1], Geo George[2], Maneesha Murali[3], Nimitha k[4], Sangeetha B[5], Blessy Rapheal[6]

[1,2,3,4,5]Btech students, Department of Electronics and Communication Engineering, Ammini College of engineering, Mankara, Palakkad, Kerala, INDIA

[6]Assistant professor, of Electronics and Communication Engineering, Ammini College of Engineering Palakkad, Kerala, INDIA

ABSTRACT

In this project, a novel reversible data hiding (RDH) algorithm is proposed for digital images. Instead of trying to keep the PSNR value high, the proposed algorithm enhances the contrast of a host image to improve its visual quality. The highest two bins in the histogram are selected for data embedding so that histogram equalization can be performed by repeating the process. The side information is embedded along with the message bits into the host image so that the original image is completely recoverable. The proposed algorithm was implemented on two sets of images to demonstrate its efficiency. To our best knowledge, it is the first algorithm that achieves image contrast enhancement by RDH. Furthermore, the evaluation results show that the visual quality can be preserved after a considerable amount of message bits have been embedded into the contrast-enhanced images, even better than three specific MATLAB functions used for image contrast enhancement.

Keywords-- Contrast enhancement, histogram modification, location map, reversible data hiding, visual quality

I. INTRODUCTION

Reversible Data Hiding (RDH) has been intensively studied in the community of signal processing. Also referred as invertible or lossless data hiding, RDH is to embed a piece of information into a host signal to generate the marked one, from which the original signal can be exactly recovered after extracting the embedded data. The technique of RDH is useful in some sensitive applications where no permanent change is allowed on the host signal. Most of the proposed algorithms are for digital images to embed invisible data or a visible watermark.

To evaluate the performance of a RDH algorithm, the hiding rate and the marked image quality are important metrics. There exists a trade-off between them because increasing the hiding rate often causes more distortion in image content. To measure the distortion, the peak signal-to-noise ratio (PSNR) value of the marked image is often calculated. Direct modification of image histogram provides less embedding capacity. In contrast, the more recent algorithms manipulate the more centrally distributed prediction errors by exploiting the correlations between neighboring pixels so that less distortion is caused by data hiding. Although the PSNR of a marked image generated with a prediction error based algorithm is kept high, the visual quality can hardly be improved because more or less distortion has been introduced by the embedding operations. For the images acquired with poor illumination, improving the visual quality is more important than keeping the PSNR value high. Moreover, contrast enhancement of medical or satellite images is desired to show the details for visual inspection. Although the PSNR value of the enhanced image is often low, the visibility of image details has been improved. To our best knowledge, there is no existing RDH algorithm that performs the task of contrast enhancement so as to improve the visual quality of host images. So in this study, we aim at inventing a new RDH algorithm to achieve the property of contrast enhancement instead of just keeping the PSNR value high.

In principle, image contrast enhancement can be achieved by histogram equalization. To perform data embedding and contrast enhancement at the same time, the proposed algorithm is performed by modifying the histogram of pixel values. Firstly, the two peaks (i.e. the highest two bins) in the histogram are found out. The bins between the peaks are unchanged while the outer bins are shifted outward so that each of the two peaks can be split into two adjacent bins. To increase the embedding capacity, the highest two bins in the modified histogram can be further chosen to be split, and so on until satisfactory contrast enhancement effect is achieved. To avoid the overflows and under- flows due to histogram

modification, the bounding pixel values are pre-processed and a location map is generated to memorize their locations. For the recovery of the original image, the location map is embedded into the host image, together with the message bits and other side information. So blind data extraction and complete recovery of the original image are both enabled. The proposed algorithm was applied to two set of images to demonstrate its efficiency. To our best knowledge, it is the first algorithm that achieves image contrast enhancement by RDH. Furthermore, the evaluation results show that the visual quality can be preserved after a considerable amount of message bits have been embedded into the contrast-enhanced images, even better than three specific MATLAB functions used for image contrast enhancement.

II. LITERATURE SURVEY

2.1 REVERSIBLE DATA EMBEDDING USING A DIFFERENCE EXPANSION

Reversible data embedding has drawn lots of interest recently. Being reversible, the original digital content can be completely restored. This is a novel reversible data embedding method for digital images. This explore the redundancy in digital images to achieve very high embedding capacity, and keep the distortion low. Reversible data embedding, which is also called lossless data embedding, embeds invisible data (which is called a payload) into a digital image in a reversible fashion. As a basic requirement, the quality degradation on the image after data embedding should be low. An intriguing feature of reversible data embedding is the reversibility, that is, one can remove the embedded data to restore the original image. From the information hiding point of view, reversible data embedding hides some information in a digital image in such a way that an authorized party could decode the hidden information and also restore the image to its original, pristine state. The motivation of reversible data embedding is distortion-free data embedding. Though imperceptible, embedding some data will inevitably change the original content. Even a very slight change in pixel values may not be desirable, especially in sensitive imagery, such as military data and medical data. In such a scenario, every bit of information is important. Any change will affect the intelligence of the image, and the access to the original, raw data is always required.

It is a high-capacity, high visual quality, reversible data-embedding method for digital images. This method can be applied to digital audio and video as well. We calculate the differences of neighboring pixel values, and select some difference values for the difference expansion (DE). The original content restoration information, a message authentication code, and additional data (which could be any data, such as date/time information, auxiliary data, etc.) will all be embedded into the difference values. In this paper we will consider grayscale images only. For color images, there are several options. One can decorrelate the dependence among different color components by a reversible color conversion transform, and then reversibly embed the data in the decorrelated components. Or one can reversibly embed each color component individually.

2.2 REVERSIBLE WATERMARK USING DIFFERENCE EXPANSION OF TRIPLETS

A new reversible watermarking algorithm based on the difference expansion of colored images has been developed. Since the watermark is completely reversible, the original image can he recovered exactly. The algorithm uses spatial and spectral triplets of pixels to hide pairs of bits: which allows the algorithm to hide a large amount of data. A spatial triplet is any three pixel values selected from the same spectral component, while a spectral triplet is any three pixel values selected from different spectral components. The algorithm is recursively applied to the rows and columns of the spectral components of the image and across all spectral components to maximize the hiding capacity Simulation results show that the hiding capacity of the algorithm is very high and the resulting distortion is low. In digital watermarking or steganography a hardly noticeable noise-like signal is usually embedded into a digital medium, such as an image audio, or video data to protect it from illicit use and alteration, to authenticate its content and origin or to enhance its value and enrich its information content. Unlike metadata, which is often appended to the digital file, a watermark is hound with the fabric of the media and cannot be removed or easily destroyed.

The watermarking process usually introduces irreversible degradation of the anginal medium. Although this degradation is slight, it may not be acceptable to some applications, such as military and medical uses, and, hence there is a need for a reversible watermark. If the embedding algorithm and all embedding parameters are available to the reader, it might be possible to calculate the watermark and subtract it from the marked medium in order to recover the original medium, once the watermark is detected and the payload is read. Unfortunately these requirements are often not available, and, furthermore, most watermarking algorithms often employ some kind of non-linearity to optimize their performance. Therefore, a reversible watermark must be designed such that it can be removed to restore the original medium without any reference to information beyond what is available in the watermarked medium itself.

2.3 REVERSIBLE DATA HIDING USING VISUAL CRYPTOGRAPHY

World witnessed the explosive growth in the communication technologies in the last decade by exploiting innovative and incredible technological developments in electronic miniaturization and efficient computational algorithms. Society have corroborated the global spread of internet, which pace accelerated the use of

exhilarated multimedia information for the better cognizance of information. Since the visual perception can exude the large amount of data at a glance, among the multimedia information digital images have gained prominence. Sometimes circumstances force user to transmit some secret information along with original image that has to be revealed after strictly verifying the authenticity of the prompting entity. Application of such a system can be explained with the following example. Suppose a new generation hospital is adopting the cloud storage technology for the biomedical images & videos (MRI, MRA and etc.),since the patient's personal information and medical data has been openly available in cloud servers, the chances of violation of privacy as well as the trespass towards confidentiality to the patient's information may result. Under these situations management can deploy the proposed algorithm as a remedy to overcome the above said problems. Proposed algorithm first encrypt the medical image of the patient with a key only known to the doctor of the patient and this encrypted version of the image only passed to assistant of doctor. At this point assistant without knowing image content embeds the personal information, disease and diagnosis information of the patient using the data hiding key known to him and doctors, thus confidentiality of image content has been protected.

This encrypted - embedded image only passes to the hospital administration or database management or systems head for storing in cloud server. Later upon specific appointment time database manager can supply encrypted-embedded image from cloud server to doctor and decoding is possible for doctor. Doctor have two options when he receives encrypted – embedded image, doctor can read only the patients information or can view the medical image alone or can perform these two in any order. This kind of facility put forwarded by the algorithm is addressed as separable manner of operation. 20 It is very clear from the example that algorithm will deploy two branches of information technology, cryptography and steganography. Cryptography for encrypting the image and steganography for the data hiding in encrypted images.

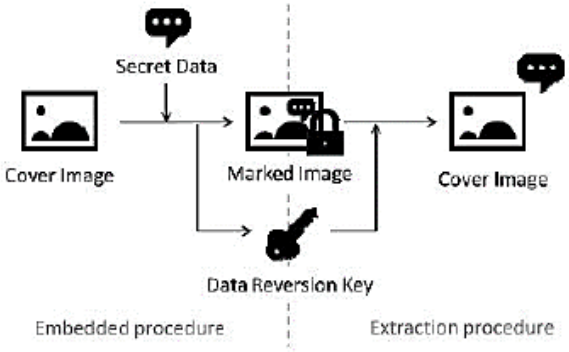

Fig 2.1 Model of the system

The system is basically divided into three different module content owner, data hider and receiver. Content owner is performing the image encryption using the data hiding to protect the confidentiality of image content. Encrypted image then passed to the data hider, with the help of a key called data hiding key data hider performs embedding of data into the encrypted image. At the receiver side there are three basic options, if the receiver is supplied only with data hiding key, only the hidden data can be extracted out not the image or any fragment of encrypted image. Second option is receiver is supplied with encryption key alone, then encrypted image with data hidden can be decrypted to the image similar to the original image but the hidden data or any fragment of hidden data cannot be extracted out. Third option is the combination of first and second options, but it can be in any order. From the discussion of option one and option two, reader can easily understand third option can be implemented in any order.

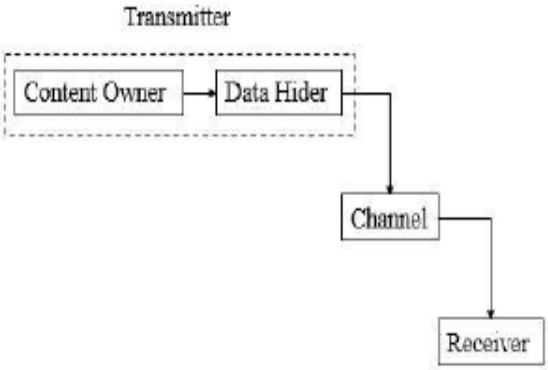

Fig 2.2 Basic entity blocks

III. PROPOSED SYSTEM

3.1 DATA EMBEDDING BY HISTOGRAM MODIFICATION

The algorithm to be presented is primarily for gray-level images but can be easily extended to color images. Given an 8-bit gray-level image, the image histogram can be calculated by counting the pixels with a gray-level value j for $j \in \{0,1, \dots \dots 254,255\}$. We use h_I to denote the image histogram so that $h_{I(j)}$ represents the number of pixels with a value. Suppose I consists of N different pixel values. Then there are N nonempty bins in $h_{I(j)}$, from which the two peaks (i.e. the highest two bins) are chosen and the corresponding smaller and bigger values are denoted by I_S and I_R, respectively. For a pixel counted in h_I with value, data embedding is performed by

$$i' = \begin{cases} i-1, & \text{for } i < I_S \\ I_S - b_k, & \text{for } i = I_S \\ i, & \text{for } I_S < i < I_R \\ I_R + b_k, & \text{for } i = I_R \\ i+1, & \text{for } i > I_R, \end{cases} \quad (1)$$

where i' is the modified pixel value, and b_k is the k-th message bit (0 or 1) to be hidden. By applying Eq (1) to every pixel counted in h_I totally $h_I(I_S)+h_I(I_R)$ binary values are embedded. Given that there is no bounding value (0 or 255) in I (otherwise pre-process is needed), there will be $N+2$ bins in the modified histogram. That is, the bins between the two peaks are unchanged while the outer ones are shifted outward so that each of the peaks can be split into two adjacent bins (i.e.I_S-1 and I_S ,I_R andI_R+1, respectively).

The peak values I_S and I_R need to be provided to extract the embedded data. One way to keep them is to exclude 16 pixels in I from histogram computing. The least significant bits (LSB) of those pixels are collected and included in the binary values to be hidden. After applying Eq. (1) to each pixel counted in h_I for data embedding, the values of I_S and I_R (each with 8 bits are used to replace the LSBs of the 16 excluded pixels by bitwise operation. To extract the embedded data, the peak values need to be retrieved and the histogram of the marked image I' is calculated excluding the 16 pixels aforementioned. Then the following operation is performed on any pixel counted in the histogram and with the value of I_S-1,I_S, I_R or I_R+1:

$$b'_k = \begin{cases} 1, & \text{if } i' = I_S - 1 \\ 0, & \text{if } i' = I_S \\ 0, & \text{if } i' = I_R \\ 1, & \text{if } i' = I_R + 1, \end{cases} \quad (2)$$

Where b'_k is the k^{th} binary value extracted from the marked image I'. The extraction operations are performed in the same order as that of the embedding operations. According to Eq. (1) the following operation is performed on every pixel counted in the histogram to recover its original value:

$$i = \begin{cases} i'+1, & \text{for } i' < I_S - 1 \\ I_S, & \text{for } i' = I_S - 1 \text{ or } i' = I_S \\ I_R, & \text{for } i' = I_R \text{ or } i' = I_R + 1 \\ i'-1, & \text{for } i' > I_R + 1 \end{cases}. \quad (3)$$

The original LSBs of 16 excluded pixels are obtained from the extracted binary values. The excluded pixels can be restored by writing them back so as to recover the original image.

3.2 PRE-PROCESS FOR COMPLETE RECOVERY

In the aforementioned algorithm, it is required that all pixels counted in h_I are within {1,……,254}. If there is any bounding pixel value (0 or 255), overflow or underflow will be caused by histogram shifting. To avoid it, the histogram needs to be pre-processed prior to the histogram modification operations. Specifically, the pixel values of 0 and 255 are modified to 1 and254, respectively. Therefore, no overflow or underflow will be caused because the possible change of each pixel value is ± 1. To memorize the pre-processed pixels, a location map with the same size as the original image is generated by assigning 1 to the location of a modified pixel, and 0 to that of an unchanged one (including the 16 excluded pixels). The location map can be pre-computed and included into the binary values to be hidden. In the extraction and recovery process, it can be obtained from the data extracted from the marked image so that the pixels modified in the pre-process can be identified. By restoring the original values of those pixels accordingly, the original image can be completely recovered.

3.3 CONTRAST ENHANCEMENT

In Section 3.1, each of the two peaks in the histogram is split into two adjacent bins with the similar or same heights because the numbers of 0s and 1s in the message bits are required to be almost equal. To increase the hiding rate, the highest two bins in the *modified* histogram are further chosen to be split by applying Eq. (1) to all pixels counted in the histogram. The same process can be repeated by splitting each of the two peaks into two adjacent bins with the similar heights to achieve the *histogram equalization* effect. In this way, data embedding and contrast enhancement are simultaneously performed. Given that the pair number of the histogram peaks to be split is , the range of pixel values from 0 to L-1 are added by while the pixels 256-L from to 255 are subtracted by L in the pre-process (noting L is a positive integer). A location map is generated by assigning 1s to the modified pixels, and 0s to the others.

The location map can be pre-computed and compressed to be firstly embedded into the host image. The value of the size of the compressed location map and the previous peak values in contrary are embedded with the last two peaks to be split, whose values are stored in the LSBs of the 16 excluded pixels. In the extraction process, the last split peak values are retrieved and the data embedded with them are extracted with Eq. (2). After restoring the histogram with Eq. (3), the data embedded with the previously split peaks can also be extracted by processing them pair by pair. At last, the location map is obtained from the extracted data to identify the pixel values modified in the pre-process.

IV. BLOCK DIAGRAM

ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

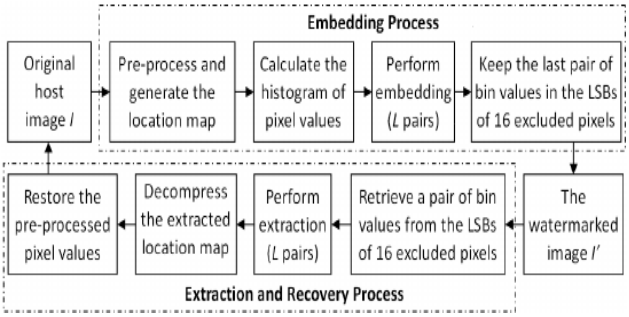

Fig 4.1 Block diagram of the proposed system

The procedure of the proposed algorithm is illustrated in Fig 4.1. Given that totally pairs of histogram bins are to be split for data embedding, the embedding procedure includes the following steps:

1) Pre-process: The pixels in the range of [0, L-1] and [256-L, 255] are processed excluding the first 16 pixels in the bottom row. A location map is generated to record the locations of those pixels and compressed by the JBIG2 standard to reduce its length.

2) The image histogram is calculated without counting the first 16 pixels in the bottom row.

3) Embedding: The two peaks (i.e. the highest two bins) in the histogram are split for data embedding by applying Eq. (1) to every pixel counted in the histogram. Then the two peaks in the *modified* histogram are chosen to be split, and so on until pairs are split. The bitstream of the compressed location map is embedded before the message bits (binary values). The value of, the length of the compressed location map, the LSBs collected from the 16 excluded pixels, and the previous peak values are embedded with the last two peaks to be split.

4) The lastly split peak values are used to replace the LSBs of the 16 excluded pixels to form the marked image.

The **extraction** and **recovery** process include the following steps:

5) The LSBs of the 16 excluded pixels are retrieved so that the values of the last two split peaks are known.

6) The data embedded with the last two split peaks are extracted by using Eq. (2) so that the value of, the length of the compressed location map, the original LSBs of 16 excluded pixels, and the previously split peak values are known. Then the recovery operations are carried out by processing all pixels except the 16 excluded ones with Eq. (3). The process of extraction and recovery is repeated until all of the split peaks are restored and the data embedded with them are extracted.

7) The compressed location map is obtained from the extracted binary values and decompressed to the original size.

8) With the decompressed map, those pixels modified in preprocess are identified. Among them, a pixel value is subtracted by if it is less than 128, or increased by otherwise. To comply with this rule, the maximum value of is 64 to avoid ambiguity. At last, the original image is recovered by writing back the original LSBs of 16 excluded pixels.

V. SIMULATION AND DISCUSSIONS

5.1 IMPLEMENTATION

REQUIREMENTS:

1. MATLAB Open Source Software
2. Image and MATLAB code in the same folder
3. Image should be .jpg , .png formats

STEPS FOR IMPLEMENTATION:

1. Open Matlab software and paste the embedding code.
2. Put the name of image which is being used in 'imread' function in the code.
3. Run the code to get the following output files:
   - Embedding Histogram
   - Image after embedding
   - Difference Image after embedding
4. Now after the embedding process paste the extraction code.
5. Take the stego image(image after embedding) and put in 'imread' function in the code.
6. Run the code to get the following output files:
   - Extraction Histogram
   - Image after extraction

5.2 FLOW CHART OF THE PROPOSED SYSTEM

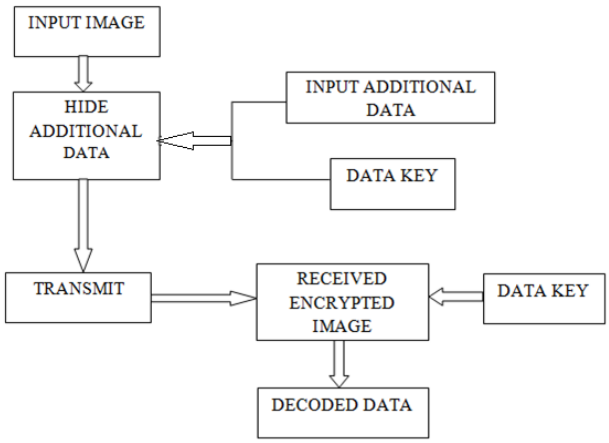

Fig 5.1 Flow chart of the proposed system

In this separable reversible data hiding in images, a content owner encrypts a secret data into the preprocessed image. A data hider may compress the least significant bits of the image using a data hiding key to create a sparse space to accommodate some additional data. With an image containing additional data, if a receiver has the data hiding key ,he can extract the additional data. Using a secret key provide additional security to the proposed system. In the first step, insert the input image in which the data to be hided. This image is preprocessed using median filter to avoid under flow and over flow. This image is further used for data hiding

ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

process. The input image is compressed with additional data and is provided with an encryption key, which is transmitted to the receiver side.

First we run the main code in MATLAB. Then we are asked to enter the secret data and also the key. Message boxes will appear to give notifications to enter key and data. Now the encryption procedure is over. This is transmitted to the destination end who knows the decryption key. If we enter any faulty key to extract the hidden data, it fails to work. Both encryption and decryption keys should be matched. The changes occurring with the image on each step of encryption and decryption are also shown in the window.

5.3 SIMULATION RESULTS

Fig 5.2 Input image

Fig 5.3 embedding data

Fig 5.4 After applying Median filter

Fig 5.5 Image with data embedded

Fig 5.6 Entering key for extraction

ISSN (ONLINE): 2250-0758, ISSN (PRINT): 2394-6962

Fig 5.7 Recovery of data

Fig 5.8 Enhanced image

VI. APPLICATIONS

a) Medical safety: Current image formats such as DICOM separate image data from the text (such as patients name, date and physician), with the result that the link between image and patient occasionally gets mangled by protocol converters. Thus embedding the patients name in the image could be a useful safety measure.
b) Terrorism: According to government officials terrorists use to hide information, maps and photographs of targets for communicating or instructing other terrorists or their alliance groups.
c) Hacking: The hacker hides a monitoring tool, server behind any image or audio or text file shares it with mail or chat which will get installed and executed. This silent installer monitoring tool will help the hacker to perform hacking within the workstation.
d) Intellectual property offenses: Intellectual property, defined as the formulas, prototypes, copyrights and customer lists maintained by a company, can be far more valuable than the actual items they sell.
e) Corporate espionage: Usage of spies to collect information about what another entity is doing or planning in a corporate environment.
f) Watermarking: Special inks to write hidden messages on bank notes and also the entertainment industry using digital watermarking and fingerprinting of audio and video for copyright protection.
g) Automatic monitoring of radio advertisements: It would be convenient to have an automated system to verify that adverts are played as contracted.

VII. CONCLUSION

A new reversible data hiding algorithm has been proposed with the property of contrast enhancement. Basically, the two peaks (i.e. the highest two bins) in the histogram are selected for data embedding so that histogram equalization can be simultaneously performed by repeating the process. The experimental results have shown that the image contrast can be enhanced by splitting a number of histogram peaks pair by pair. Compared with the special MATLAB functions, the visual quality of the contrast-enhanced images generated by our algorithm is better preserved. Moreover, the original image can be exactly recovered without any additional information. Hence the proposed algorithm has made the image contrast enhancement reversible. Improving the algorithm robustness, and applying it to the medical and satellite images for the better visibility, will be our future work.

REFERENCES

[1] Hao-Tian Wu, *Member, IEEE*, Jean-Luc Dugelay, *Fellow, IEEE*, and Yun-Qing Shi, *Fellow, IEEE*"Reversible Image Data Hiding with Contrast Enhancement".
[2] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett*, vol. 14, no. 4, pp. 255–258, Apr. 2007.
[3] D.M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
[4] H. T.Wu and J. Huang, "Reversible image watermarking on prediction error by efficient histogram modification," *Signal Process.*, vol. 92, no. 12, pp. 3000–3009, Dec. 2012.

[5] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
[6] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, May 2000.
[7] M.-Z.Gao, Z.-G.Wu, and L.Wang, "Comprehensive evaluation forHE based contrast enhancement techniques," *Adv. Intell. Syst. Applicat.*, vol. 2, pp. 331–338, 2013.
[8] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans.Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.