IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 July 2023, accepted 23 July 2023, date of publication 1 August 2023, date of current version 7 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3300918

RESEARCH ARTICLE

Digital Image Steganography With Error Correction on Extracted Data

S. N. V. J. DEVI KOSURU[1], ANITA PRADHAN[1], K. ABDUL BASITH[2], RESHMA SONAR[3], AND GANDHARBA SWAIN[4]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh 522302, India
[2]Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology and Management, Hyderabad 500043, India
[3]Department of Artificial Intelligence and Machine Learning, ISBM College of Engineering, Pune, Maharashtra 412115, India
[4]Department of Artificial Intelligence and Data Science, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh 522302, India

Corresponding author: Gandharba Swain (gswain1234@gmail.com)

This work was supported by the Koneru Lakshmaiah Education Foundation (KLEF) under Grant KLEF-2002030003.

ABSTRACT The hiding capacity (HC), imperceptibility, and security are the 3 important quality measures for a steganography technique. While the stego-image is on transit on the internet, the hidden data may be changed because of various reasons. The existing techniques does neither focus on detecting the errors in the data nor to correct the errors in data. Therefore, this article brings forward a steganography technique, wherein error detection and correction can be performed at recipient side. The original image is logically sliced into 2×2 disjoint blocks. From these 4 pixels, 4 quotients and 4 least significant bits (LSBs) are generated. Each quotient is the decimal value of 7 most significant bits (MSBs) of a pixel. In every block 8 data bits can be camouflaged. From the 8 data bits, 4 redundant bits are computed using modified Hamming code. The 8 data bits and one redundant bit are camouflaged in the quotients by either quotient value differencing (QVD) or bit substitution. If camouflaging is performed in quotients using QVD, then indicator bit is set to1. Otherwise, if camouflaging in quotients is performed using bit substitution, then indicator bit is set to 0. The 3 remaining redundant bits along with the indicator bit are stored in the LSBs of the 4 pixels. At the receiver side, data could be extracted, and error correction procedure could be applied to correct 1-bit error over the 8 bits of data extracted from a block. From the experimental reports it could be concluded that the errors in the retrieved data at the recipient can be detected and corrected without reducing the HC and without increasing the distortion.

INDEX TERMS Data hiding, steganography, error correction, QVD, modified Hamming code.

I. INTRODUCTION

In an image steganography technique, the classified data is camouflaged inside an image in such a manner that the visual and statistical properties are preserved. We have 2 traditional approaches for image steganography, (i) Least significant bit (LSB) substitution, and (ii) pixel value differencing (PVD). In LSB substitution steganography the LSB of each pixel can be replaced by a secret data bit to hide the secret information throughout the image. The hiding capacity (HC) will be only one bit per pixel. If we want to hide a greater number of bits, we can extend the substitution up to 2 or 3 LSBs. In rare cases to hide a very large number of bits we may extend this concept up to 4 LSBs. But if we go up to 4 LSBs then distortion will be high, which can be easily detected by various detection mechanisms. The LSB replacement is very simple and detectable by regular-singular (RS) analysis [1]. RS analysis is a steganalysis mechanism, which successfully detects the LSB substitution. The PVD approach was initiated by Wu and Tsai [2]. As per this approach the image is partitioned into non-overlapped blocks, and each block contains 2 pixels. The difference between these two pixels is computed. If the difference value is low, then the block falls in a smoother region of the image, so lesser number of bits shall be hidden in this block. If the difference value is larger, then the block belongs to a textured region, so that we can hide a greater

The associate editor coordinating the review of this manuscript and approving it for publication was Zeev Zalevsky.

number of bits without sacrificing imperceptibility. The PVD technique is undetected by RS analysis but detected by pixel difference histogram (PDH) analysis [3]. Here the difference in two-pixel values is computed to measure the HC, and then changed by a new difference value to hide data. Later, more efficient PVD based techniques came into existence. Lee et al. [4] came up with PVD in 2 × 2 blocks to acquire higher HC. Darabkh et al. [5] integrated LSB substitution and PVD with large size blocks to improve the security and HC. Mukherjee et al. [6] too developed a PVD to store data bits in randomized positions. They targeted both smooth and edge regions. They claimed that it protects a variety of steganalysis attacks.

As per Wu et al.'s view if we use PVD approach in edge areas and LSB alteration in smooth areas, then peak signal-to-noise ratio (PSNR) and HC can be boosted to a greater extent [7]. There are a good number of variants of PVD steganography, these are described in related work section. If a steganography technique uses any improved version of LSB substitution, then its security shall be tested by RS analysis. Similarly, if any improved version of PVD technique is developed, then its security shall be tested by performing PDH analysis. Furthermore, if an improved version of steganography uses both LSB approach and PVD approach, then its security shall be tested by performing both RS analysis and PDH analysis.

There exist numerous applications of data hiding including safe transfer of healthcare data [29], and secretly storing of data in images of social network websites [30]. While proposing any new steganography technique, the authors try to improve at least one of the three quality parameters, (i) HC, (ii) imperceptibility, and (iii) security [31], [32]. Rustad et al. [32] achieved higher imperceptibility by embedding the data bits at a possible bit pattern with minimum distortion. Of course, to search the minimum distortion pattern it increased the embedding time but achieved better imperceptibility. Liao et al. [33] proposed another interesting application of steganography for storing data in cloud securely with multiple images. As per this approach different parts of secret message are camouflaged in multiple images. Based on the texture characteristics of images, variable amount of payload can be embedded. If the distribution strategy is proper, then better security can be achieved. In an RGB image there are 3 colour channels. Authors in [34] improved the imperceptibility in RGB image steganography by introducing inter-channel relationship for data hiding. The HC of a channel depends on the other 2 channels in the pixel. It has been believed that for better imperceptibility all the 3 channels should either be increased or be decreased after hiding the bits.

II. RELATED WORK AND AUTHORS' CONTRIBUTIONS

A. RELATED WORK

Pradhan et al. [8] proposed a PVD technique using 3 × 3 size blocks to improve the hiding capacity [8]. In this case 8 possible directions are used to compute the differences and these differences are altered to hide secret bits. Authors in [9] did PVD using add or subtract operation. In their approach a block of 3 consecutive pixels is taken, LSB substitution is done in middle pixel. After the middle pixel value is changed, 2 difference values are computed with the 2 neighbors and addition-subtraction based logic is applied to hide bits in them. Although the HC is improved, the fall off boundary problem (FOBP) arose and PDH test could catch it. Swain [10] extended the idea of addition or subtraction to larger size blocks to protect from PDH analysis and improve upon HC and PSNR. Shukla et al. [11] also used the idea of addition or subtraction with compression and encryption, so that they could improve upon security and HC. Authors in [12] plied histogram-based concept with LSB and PVD to increase HC, and PSNR. Authors in [13] categorized blocks into 3 classes, (i) more complex, (ii) less complex, and (iii) smooth. They plied PVD on more complex and less complex blocks, and LSB on smooth blocks. Although improvement was brought in HC and PSNR, but it converges to Wu et al.'s approach.

The concept of LSB substitution with QVD in 2-bit planes was originated by Jung [14]. He logically sliced an image into 1 × 2 size non-overlapping pixel blocks. Consider (P_1, P_2) as one block. From these two pixels, two quotients (Q_1, Q_2) and two remainders (R_1, R_2) are derived. Q_1 is decimal number for (8-k) MSBs of P_1, R_1 is decimal equivalent of k LSBs of P_1, Q_2 is decimal number for (8-k) MSBs of P_2, and R_2 is the decimal equivalent of k LSBs of P_2. The traditional PVD approach of Wu & Tsai can be plied to camouflage data in 2 quotients and LSB substitution can be plied to store data in 2 remainders. Although this approach attained a higher HC but suffered with FOBP and incorrect extraction problem (IEP). Referring to addition-subtraction based PVD of Khodaei & Faez, Pradhan et al. [15] did addition-subtraction based quotient value differencing (ASQVD) to boost upon the HC. They coupled this addition subtraction based QVD with neighbors' match. In this mechanism 3 × 3 blocks are used, and data camouflaging is done in 2 steps. In the first step, QVD and LSB concept is applied on 5 pixels. Thereafter, neighbors match approach is applied on remaining 4 pixels. This scheme possesses lesser PSNR and HC. Furthermore, Swain [16] used 3 × 3 magnitude blocks for QVD to increase the HC. Further, Liu et al. [17] integrated PVD with the approach of neighbors' match in 3 × 3 magnitude blocks. They hid data plying PVD on middle pixel and 4 surrounding pixels and hid data in remaining 4 corner pixels by neighbors' match approach. By using this hybrid approach, they increased the HC, but also increased the time complexity. Singh [18] also followed Jung's PVD+LSB approach with different magnitude blocks, so that HC was improved. Sonar and Swain [19] advanced this idea by combining pixel value correlation (PVC) with QVD. In this approach an image is sliced into 3 × 3 size blocks. Out of 9 pixels in the block, camouflaging is done first in 5 pixels using QVD and LSB approach. After that, the PVC approach is applied

to hide data in rest of the pixels. They achieved higher HC and PSNR along with protection from security attacks. Khadse and Swain [20] proposed QVD with LSB substitution addressing the IEP that arose in Jung's technique. Swain and Pradhan [21] also proposed a hybrid approach using QVD and quotient value correlation (QVC) with data integrity verification at the receiver.

Wang et al. [22] introduced modulus function (MF) and PVD in 1×2 magnitude blocks. The HC of a block is determined by remainder value obtained from the MF. Zhao et al. [23] advanced this MF based idea using some optimized equations to attain higher PSNR value. Swain [24] recognized a range-mismatch in scheme of [22] and made a superior design to avoid it using 2×3 magnitude blocks. It not only avoided the range mismatch problem, but also provided attack resistance, higher HC and higher PSNR. Maniriho and Ahmad [25] proposed the concept of difference expansion with MF, but it did not give better HC. Further, Li and He [26] introduced particle swarm optimization (PSO) along with PVD plus MF to attain higher PSNR value and good quality stego-image (SI). While hiding data in 3 components of a color image pixel, if we treat them distinctly, then the inherent statistical correlation among the components will be disarranged. To stop this situation, authors in [27] suggested that while hiding the data the 3 components' values must be either increased or decreased together. Li et al. [28] too said that in edge areas, all the pixel values must be either increased or decreased to attain greater security.

B. RESEARCH CONTRIBUTIONS

Traditional image steganography techniques aim at three major parameters, (i) HC, (ii) imperceptibility, and (iii) security. But while the image is in transit in an unsecured medium, there is a chance that some of the hidden data bits will be changed. So, we shall check the errors and correct the error bits in the extracted data. This is a very important problem. This article brings up a steganography technique for identifying the errors in the retrieved data and for correcting the identified error bits. However, the HC and PSNR are not sacrificed.

The image is sliced into 2×2 blocks. In a block, 8 data bits and 4 redundant bits are camouflaged by using QVD and LSB substitution. The redundant bits are calculated from the data bits by introducing modified Hamming code (MHC), so that at receiver side the error bits positions can be accurately identified. Figure 1 represents the arrangement of data and redundant bits in (12, 4) Hamming code. There are 12 bits, out of which 8 are data bits and 4 are redundant bits. The data bits are denoted as D and the redundant bits are denoted as R_1, R_2, R_3, and R_4. R_1 is the parity over the bits at places 1,3,5,7,9, and 11. R_2 is the parity over the bits at places 2,3,6,7,10, and 11. R_3 is the parity over the bits at places 4,5,6,7, and 12. Similarly, R_4 is the parity over the bits at places 8,9,10,11, and 12.

12	11	10	9	8	7	6	5	4	3	2	1
D	D	D	D	R_4	D	D	D	R_3	D	R_2	R_1

FIGURE 1. The data and redundant bits in HC.

12	11	10	9	8	7	6	5	4	3	2	1
D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	R_4	R_3	R_2	R_1

FIGURE 2. The data and redundant bits in MHC.

The proposed MHC is shown in Fig.2, and the redundant bits are computed in Eqs.1 and 2. Here R_1 is the parity over the bits at places 5,6,8,9, and 11. R_2 is the parity over the bits at places 5,7,8,10, and 11. R_3 is the parity over the bits at places 6,7,8, and 12. Similarly, R_4 is the parity over the bits at places 9,10,11, and 12.

$$R_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7, \textit{ and}$$
$$R_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 \quad (1)$$
$$R_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8, \textit{ and}$$
$$R_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8 \quad (2)$$

III. PROPOSED QVD+MHC METHODOLOGY

The flow diagrams for embedding and extraction procedures of the proposed QVD+MHC methodology are depicted in Fig.3, and Fig.4 respectively. The detailed step-by-step procedures of embedding and extraction are illustrated in section III-A and section III-B respectively.

A. THE DATA EMBEDDING PROCEDURE

From the original image (OI), 2×2 size blocks are created in disjoint manner. Suppose Fig.5(a) is a sample block wherein the 4 pixels are P_x, P_1, P_2 and P_3. The data camouflaging procedure is described below.

Step 1: Divide every pixel of a block into 2 parts. The decimal value of first 7 MSBs of P_x is known as Q_x and the LSB bit is known as L_x. The decimal value of first 7 MSBs of P_1 is known as Q_1 and LSB bit is known as L_1. The decimal value of first 7 MSBs of P_2 is known as Q_2 and LSB bit is known as L_2. The decimal value of first 7 MSBs of P_3 is known as Q_3 and LSB bit is known as L_3. These 4 quotients are shown as a quotient block in Fig.5(b) and the LSBs of 4 pixels are shown as a LSB block in Fig.5(c). Furthermore, Fig.6 gives a bit level illustration.

In fact, the quotients and LSBs are computed using (3) and (4) accordingly, where div is the quotient division and mod is the remainder division.

$$Q_x = P_x \textit{ div } 2, Q_1 = P_1 \textit{ div } 2, Q_2 = P_2 \textit{ div } 2, \textit{ and}$$
$$Q_3 = P_3 \textit{ div } 2 \quad (3)$$
$$L_x = P_x \textit{ mod } 2, L_1 = P_1 \textit{ mod } 2, L_2 = P_2 \textit{ mod } 2, \textit{ and}$$
$$L_3 = P_3 \textit{ mod } 2 \quad (4)$$

Step 2: Take next 8 bits from secret binary data stream and denote them as $D_8D_7D_6D_5D_4D_3D_2D_1$. Compute 4 redundant bits R_4, R_3, R_2, R_1 as per the proposed MHC by (1) and (2).

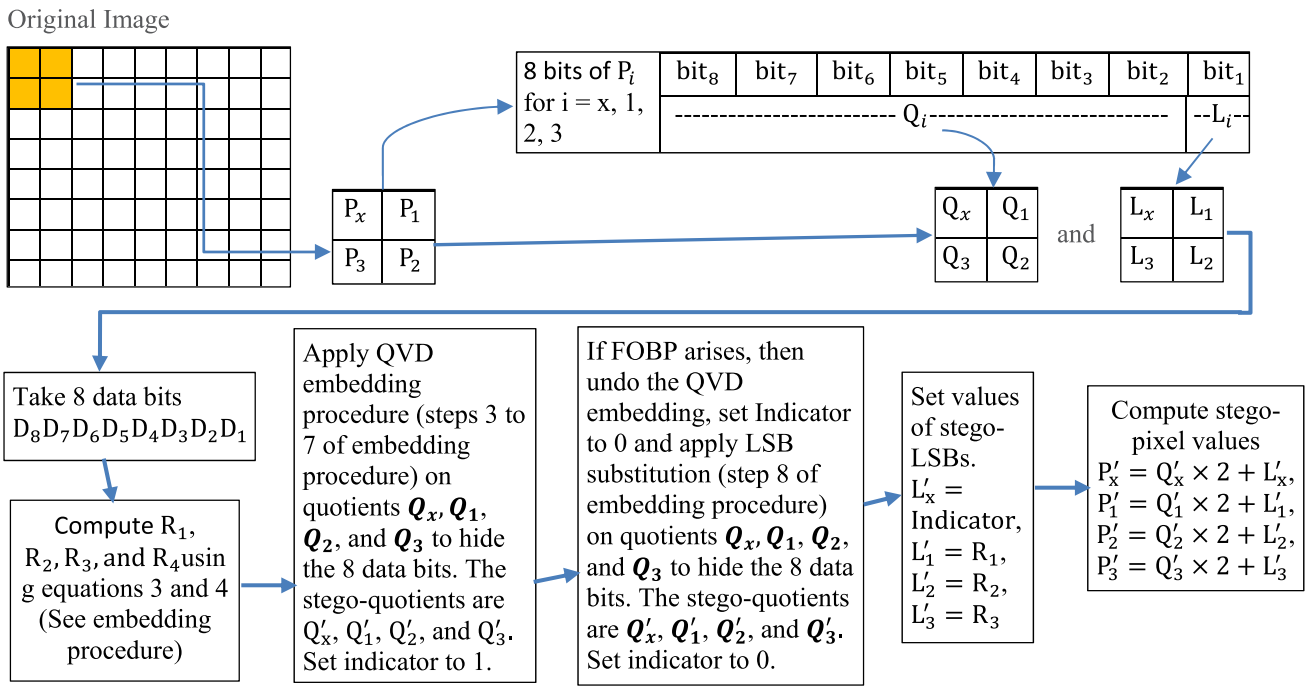


FIGURE 3. A flow diagram of data embedding procedure.

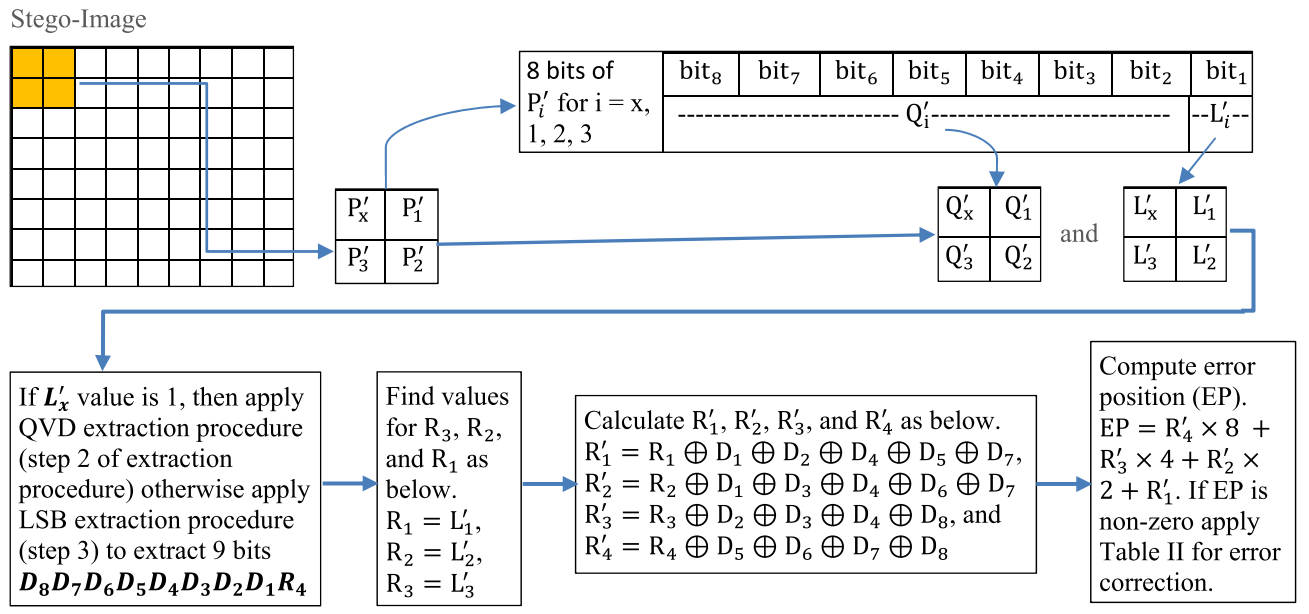


FIGURE 4. A flow diagram of data extraction procedure.

Figure 2 represents the positions of various data and redundant bits. Represent the 3 bits $D_8D_7D_6$ in decimal value b_1. Similarly, represent $D_5D_4D_3$ in decimal value b_2. Represent $D_2D_1R_4$ in decimal value b_3.

Step 3:Compute 3 difference values d_1, d_2 and d_3 as in (5).

$$d_1 = (Q_x - Q_1), d_2 = (Q_x - Q_2), d_3 = (Q_x - Q_3) \quad (5)$$

The absolute values of these difference values fall in one of the 16 quantization ranges (QR) of Table 1. Here, the lower bound (LB) and the upper bound (UB) of the ranges are specified in 2nd and 3rd rows respectively. Suppose d_1 falls in a range and its LB is denoted as LB_1. Similarly, d_2 falls in a range, its LB is denoted as LB_2 and d_3 falls in a range, its LB is denoted as LB_3.

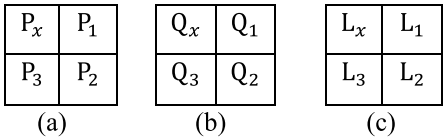


FIGURE 5. (a) Pixel block, (b) Quotient block and (c) LSB block.

TABLE 1. The range table.

Quantization Ranges (QRs)	QR 1	QR 2	QR 3	QR 4	QR 5	QR 6	QR 7	QR 8	QR 9	QR 10	QR 11	QR 12	QR 13	QR 14	QR 15	QR 16
LB	0	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120
UB	7	15	23	31	39	47	55	63	71	79	87	95	103	111	119	127

P_x	bit_8	bit_7	bit_6	bit_5	bit_4	bit_3	bit_2	bit_1
	---------- Q_x ----------							--L_x--
P_1	bit_8	bit_7	bit_6	bit_5	bit_4	bit_3	bit_2	bit_1
	---------- Q_1 ----------							--L_1--
P_2	bit_8	bit_7	bit_6	bit_5	bit_4	bit_3	bit_2	bit_1
	---------- Q_2 ----------							--L_2--
P_3	bit_8	bit_7	bit_6	bit_5	bit_4	bit_3	bit_2	bit_1
	---------- Q_3 ----------							--L_3--

FIGURE 6. Representation of quotients and LSBs in bit level.

Step 4: Compute 3 new difference values d'_1, d'_2 and d'_3 plying (6). Further compute m_1, m_2, and m_3 using (7).

$$d'_1 = \begin{cases} LB_1 + b_1, \text{if } d_1 \geq 0, \\ -LB_1 - b_1, \text{if } d_1 < 0, \end{cases}$$

$$d'_2 = \begin{cases} LB_2 + b_2, \text{if } d_2 \geq 0, \\ -LB_2 - b_2, \text{if } d_2 < 0 \end{cases}, \textit{and}$$

$$d'_3 = \begin{cases} LB_3 + b_3, \text{if } d_3 \geq 0, \\ -LB_3 - b_3, \text{if } d_3 < 0 \end{cases} \quad (6)$$

$$m_1 = \left|d'_1 - d_1\right|, m_2 = \left|d'_2 - d_2\right|, \textit{and } m_3 = \left|d'_3 - d_3\right| \quad (7)$$

Step 5: Form 3 quotient pairs (Q_x, Q_1), (Q_x, Q_2), and (Q_x, Q_3). The value b_1 can be hidden in pair (Q_x, Q_1) using (8), as shown at the bottom of page 7, to obtain the stego-values (Q'_x, Q'_1). The value b_2 can be hidden in pair (Q_x, Q_2) using (9), as shown at the bottom of page 7, to obtain the stego-values (Q'_x, Q'_2). Similarly, the value b_3 can be hidden in pair (Q_x, Q_3) using (10), as shown at the bottom of page 7, to obtain the stego-values (Q'_x, Q'_3). Here, the functions "ceiling" and "floor" stand for roundup to next higher and next lower integers respectively.

Step 6:Q'_x, the stego-value of Q_x, shall be selected out of the four values Q'_m, Q'_{x1}, Q'_{x2}, and Q'_{x3}, where Q'_m is calculated using (11).

$$Q'_m = \text{ceiling}\left(\frac{(Q'_{x1} + Q'_{x2} + Q'_{x3})}{3}\right) \quad (11)$$

Case 1: If we opt $Q'_x = Q'_m$, then we can set $Q'_1 = Q'_1 + (Q'_m - Q'_{x1})$, $Q'_2 = Q'_2 + (Q'_m - Q'_{x2})$, and $Q'_3 = Q'_3 + (Q'_m - Q'_{x3})$.

Case 2: If we choose $Q'_x = Q'_{x1}$, then we can set $Q'_1 = Q'_1$, $Q'_2 = Q'_2 + (Q'_{x1} - Q'_{x2})$, and $Q'_3 = Q'_3 + (Q'_{x1} - Q'_{x3})$.

Case 3: If we choose $Q'_x = Q'_{x2}$, then we can set $Q'_1 = Q'_1 + (Q'_{x2} - Q'_{x1})$, $Q'_2 = Q'_2$, and $Q'_3 = Q'_3 + (Q'_{x2} - Q'_{x3})$.

Case 4: If we choose $Q'_x = Q'_{x3}$, then we can set $Q'_1 = Q'_1 + (Q'_{x3} - Q'_{x1})$, $Q'_2 = Q'_2 + (Q'_{x3} - Q'_{x2})$, and $Q'_3 = Q'_3$.

The above 4 cases shall be explored. The one which possesses lowest mean square error (MSE) is to be chosen. The MSE is computed plying (12).

$$MSE = \left|\frac{(Q'_x - Q_x)^2 + (Q'_1 - Q_1)^2 + (Q'_2 - Q_2)^2 + (Q'_3 - Q_3)^2}{4}\right| \quad (12)$$

Step 7: After computing Q'_x, Q'_1, Q'_2, and Q'_3 as above, FOBP arises if any of these computed stego-quotient values are not in between 0 and 127. If FOBP does not arise, then set Indicator=1. Now we shall find L'_x, L'_1, L'_2 and L'_3, the stego-values of L_x, L_1, L_2, and L_3 respectively using (13).

$$L'_x = \text{Indicator}, L'_1 = R_1, L'_2 = R_2, \textit{and } L'_3 = R_3 \quad (13)$$

Step 8:If FOBP arises, then undo the steps 3 to 7, and apply LSB substitution to camouflage the data bits and redundant bits as follows. Set Indicator=0. Hide D_8D_7 in 2 LSBs of Q_x. Hide D_6D_5 in 2 LSBs of Q_1. Hide D_4D_3 in 2 LSBs of Q_2, and hide $D_2D_1R_4$ in 3 LSBs of Q_3. After doing this, let the stego-values of the quotients be denoted as Q'_x, Q'_1, Q'_2, and Q'_3 accordingly. Furthermore, find L'_x, L'_1, L'_2 and L'_3, the stego-values of L_x, L_1, L_2, and L_3 respectively using Eq.13.

Step 9: Now compute the stego-pixel values using (14).

$$P'_x = Q'_x \times 2 + L'_x, P'_1 = Q'_1 \times 2 + L'_1,$$
$$P'_2 = Q'_2 \times 2 + L'_2, \textit{and } P'_3 = Q'_3 \times 2 + L'_3 \quad (14)$$

For ease of understanding an example of embedding procedure is depicted in Fig.7 step-by-step.

B. DATA EXTRACTION AND ERROR CORRECTION PROCEDURE

From the stego-image (SI), 2 × 2 size blocks are created in disjoint manner. Fig.8(a) is a sample block where the 4 stego-pixels are P'_x, P'_1, P'_2, and P'_3. The data retrieval and error correction are performed by the steps below.

Step 1: Compute the quotients and LSBs using (15) and (16) respectively. The stego-quotient, and stego-LSB

Example of Embedding

Step 1

Pixel block

P_x =98	P_1 =100
P_3 =101	P_2 =102

Pixel block in binary

01100010	01100100
01100101	01100110

Quotient block

Q_x=49	Q_1=50
Q_3=50	Q_2=51

L-block

L_x=0	L_1=0
L_3=1	L_2=0

Step 2

Data bits

$D_8D_7D_6D_5D_4D_3D_2D_1$
=1 0 0 1 1 0 0 1

$R_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$
$= 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$

$R_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$
$= 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$

$R_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
$= 0 \oplus 0 \oplus 1 \oplus 1 = 0$

$R_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$
$= 1 \oplus 0 \oplus 0 \oplus 1 = 0$

$b_1 = D_8D_7D_6 = 100_2$

$b_2 = D_5D_4D_3 = 110_2$

$b_3 = D_2D_1R_4 = 010_2$

$b_1 = 4, b_2 = 6, b_3 = 2$

Step 3

$d_1 = (Q_x - Q_1) = (49 - 50) = -1$
$|d_1|$ belongs to QR1, so $LB_1 = 0$

$d_2 = (Q_x - Q_2) = (49 - 51) = -2$
$|d_2|$ belongs to QR1, so $LB_2 = 0$

$d_3 = (Q_x - Q_3) = (49 - 50) = -1$
$|d_3|$ belongs to QR1, so $LB_3 = 0$

Step 4

$d_1 < 0$, so $d_1' = -LB_1 - b_1 = -0 - 4 = -4$	$d_2 < 0$, so $d_2' = -LB_2 - b_2 = -0 - 6 = -6$	$d_3 < 0$, so $d_3' = -LB_3 - b_3 = -0 - 2 = -2$
$m_1 = \|d_1' - d_1\| = \|-4 - (-1)\| = 3$	$m_2 = \|d_2' - d_2\| = \|-6 - (-2)\| = 4$	$m_3 = \|d_3' - d_3\| = \|-2 - (-1)\| = 1$

Step 5

d_1 is odd, so $Q'_{x1} = Q_x - \text{ceiling}\left(\frac{m_1}{2}\right) = 49 - \text{ceiling}\left(\frac{3}{2}\right) = 47$, and $Q_1' = Q_1 + \text{floor}\left(\frac{m_1}{2}\right) = 50 + \text{floor}\left(\frac{3}{2}\right) = 51$
d_2 is even, so $Q'_{x2} = Q_x - \text{floor}\left(\frac{m_2}{2}\right) = 49 - \text{floor}\left(\frac{4}{2}\right) = 47$, and $Q_2' = Q_2 + \text{ceiling}\left(\frac{m_2}{2}\right) = 51 + \text{ceiling}\left(\frac{4}{2}\right) = 53$
d_3 is even, so $Q'_{x3} = Q_x - \text{ceiling}\left(\frac{m_3}{2}\right) = 49 - \text{ceiling}\left(\frac{1}{2}\right) = 48$, and $Q_3' = Q_3 + \text{floor}\left(\frac{m_3}{2}\right) = 50 + \text{floor}\left(\frac{1}{2}\right) = 50$

Step 6

$$Q'_m = \text{ceiling}\left(\frac{(Q'_{x1}+Q'_{x2}+Q'_{x3})}{3}\right) = \text{ceiling}\left(\frac{(47+47+48)}{3}\right) = 48$$

Case 1: $Q'_x = Q'_m = 48$, we set $Q_1' = Q_1' + (Q'_m - Q'_{x1}) = 52$, $Q_2' = Q_2' + (Q'_m - Q'_{x2}) = 54$, and $Q_3' = Q_3' + (Q'_m - Q'_{x3}) = 50$
Case 2: $Q'_x = Q'_{x1} = 47$, we set $Q_1' = 51$, $Q_2' = Q_2' + (Q'_{x1} - Q'_{x2}) = 53$, and $Q_3' = Q_3' + (Q'_{x1} - Q'_{x3}) = 49$
Case 3: $Q'_x = Q'_{x2} = 47$, we set $Q_1' = Q_1' + (Q'_{x2} - Q'_{x1}) = 51$, $Q_2' = 53$, and $Q_3' = Q_3' + (Q'_{x2} - Q'_{x3}) = 49$
Case 4: $Q'_x = Q'_{x3} = 48$, we set $Q_1' = Q_1' + (Q'_{x3} - Q'_{x1}) = 52$, $Q_2' = Q_2' + (Q'_{x3} - Q'_{x2}) = 54$, and $Q_3' = 50$

In case 1, MSE= 3.5, in case 2, MSE= 2.5, in case 3, MSE= 2.5, and in case 4, MSE= 3.5.
Case 2, and case 3 have equal MSE, lesser than other 2 cases, so we can choose any one of these 2 cases.
Suppose we take case 2, then we make $Q'_x = 47$. Further we set $Q_1' = 51$, $Q_2' = 53$, and $Q_3' = 49$.

Step 7

The values of Q'_x, Q_1', Q_2', and Q_3' are in between 0 and 127. FOBP does not arise, so Indicator=1.
$L'_x = \text{Indicator} = 1$, $L_1' = R_1 = 1$, $L_2' = R_2 = 0$, and $L_3' = R_3 = 0$.

Step 8

In step 6 we can see that $Q'_x = 47$, $Q_1' = 51$, $Q_2' = 53$, and $Q_3' = 49$. These values are in between 0 and 127. FOBP does not arise, so no need to undo the QVD embedding (steps 3 to 7). Hence LSB substitution is not necessary.

Step 9 Stego-pixel block is computed below

$P'_x = Q'_x \times 2 + L'_x = 95$	$P_1' = Q_1' \times 2 + L_1' = 103$
$P_3' = Q_3' \times 2 + L_3' = 98$	$P_2' = Q_2' \times 2 + L_2' = 106$

FIGURE 7. The example of embedding.

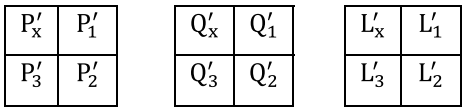


FIGURE 8. (a) Stego-pixel block, (b) Stego-quotient block, and (c) Stego-LSB block.

blocks are shown in Fig.8(b) and (c) accordingly.

$$Q'_x = P'_x \; div \; 2, Q'_1 = P'_1 \; div \; 2, Q'_2 = P'_2 \; div \; 2, Q'_3 = P'_3 \; div \; 2 \tag{15}$$

$$L'_x = P'_x \; mod \; 2, L'_1 = P'_1 \; mod \; 2, L'_2 = P'_2 \; mod \; 2, L'_3 = P'_3 \; mod \; 2 \tag{16}$$

Step 2: If $L'_x = 1$, then apply QVD extraction as follows. Compute 3 difference values d_1, d_2, and $d_3$using (17).

$$d_1 = \left|Q'_x - Q'_1\right|, d_2 = \left|Q'_x - Q'_2\right|, \; and \; d_3 = \left|Q'_x - Q'_3\right| \tag{17}$$

Suppose d_1 falls in a range of Table 1 and its LB is denoted as LB_1. Similarly, d_2 falls in a range, its LB is denoted as LB_2 and d_3 falls in a range, its LB is denoted as LB_3. Now compute b_1, b_2, and b_3 using (18).

$$b_1 = d_1 - LB_1, b_2 = d_2 - LB_2, \; and \; b_3 = d_3 - LB_3 \tag{18}$$

Convert b_1 to 3 binary bits and denote them as $D_8D_7D_6$. Convert b_2 to 3 binary bits and denote them as $D_5D_4D_3$. Similarly, convert b_3 to 3 binary bits and denote them as $D_2D_1R_4$. Find R_3, R_2, and R_1 using (19).

$$R_1 = L'_1, R_2 = L'_2, \; and \; R_3 = L'_3 \tag{19}$$

Step 3: If $L'_x = 0$, then apply LSB extraction as follows. Extract 2 LSBs from Q'_x, and denote them as D_8D_7. Extract 2 LSBs from Q'_1, and denote them as D_6D_5. Extract 2 LSBs from Q'_2, and denote them as D_4D_3. Similarly, extract 3 LSBs from Q'_3, and denote them as $D_2D_1R_4$. Find R_3, R_2, and R_1 using Eq.19.

Step 4: Either by step 2 or by step 3, we obtained the 8 data bits $D_8, D_7, D_6, D_5, D_4, D_3, D_2, D_1$and 4 redundant bits R_4, R_3, R_2, R_1. Now calculate R'_1, R'_2, R'_3, and R'_4 using (20) and (21).

TABLE 2. Error position detection and correction.

EP value	Error position	Error correction
1	Error at bit D_1	Invert the bit D_1
2	Error at bit D_2	Invert the bit D_2
3	Error at bit D_3	Invert the bit D_3
4	Error at bit D_4	Invert the bit D_4
5	Error at bit D_5	Invert the bit D_5
6	Error at bit D_6	Invert the bit D_6
7	Error at bit D_7	Invert the bit D_7
8	Error at bit D_8	Invert the bit D_8

$$R'_1 = R_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7, \; and$$
$$R'_2 = R_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 \tag{20}$$
$$R'_3 = R_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8, \; and$$
$$R'_4 = R_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 \tag{21}$$

Step 5: Compute the error position (EP) using (22). If EP=0, then there is no error and extracted binary bits are $D_8, D_7, D_6, D_5, D_4, D_3, D_2, D_1$. Otherwise, use Table 2 to correct the error. Note that only 1 bit error can be detected and corrected.

$$EP = R'_4 \times 8 + R'_3 \times 4 + R'_2 \times 2 + R'_1 \tag{22}$$

In the embedding procedure we apply either QVD or LSB substitution. If L'_x =0, we have applied LSB substitution during embedding. If L'_x =1, we have applied QVD procedure during embedding. We can correct the errors in both cases. If L'_x =0, then extract the 2 LSBs from LSBs of Q'_x, and denote them as D_8D_7. Extract 2 LSBs of Q'_1, and denote them as D_6D_5. Extract 2 LSBs of Q'_2, and denote them as D_4D_3. Similarly, extract 3 LSBs of Q'_3, and denote them as $D_2D_1R_4$. If L'_x =1, then extract the bits $D_8D_7D_6$ from (Q'_x, Q'_1) by QVD extraction procedure. Similarly, extract the bits $D_5D_4D_3$ from (Q'_x, Q'_2), and bits $D_2D_1R_4$ from (Q'_x, Q'_3). Now we set $R_1 = L'_1$, $R_2 = L'_2$, and $R_3 = L'_3$. Furthermore, calculate R'_1, R'_2, R'_3, and R'_4 using (20) and (21). If R'_1, R'_2, R'_3, and R'_4 values are all zeros, then compute EP value using (22),

$$(Q'_{x1}, Q'_1) = \begin{cases} (Q_x - \text{floor}\left(\frac{m_1}{2}\right), Q_1 + \text{ceiling}\left(\frac{m_1}{2}\right), \text{if } d_1 \text{ is even}, \\ (Q_x - \text{ceiling}\left(\frac{m_1}{2}\right), Q_1 + \text{floor}\left(\frac{m_1}{2}\right), \text{if } d_1 \text{ is odd} \end{cases} \tag{8}$$

$$(Q'_{x2}, Q'_2) = \begin{cases} (Q_x - \text{floor}\left(\frac{m_2}{2}\right), Q_2 + \text{ceiling}\left(\frac{m_2}{2}\right), \text{if } d_2 \text{ is even}, \\ (Q_x - \text{ceiling}\left(\frac{m_1}{2}\right), Q_2 + \text{floor}\left(\frac{m_2}{2}\right), \text{if } d_2 \text{ is odd} \end{cases} \tag{9}$$

$$(Q'_{x3}, Q'_3) = \begin{cases} (Q_x - \text{floor}\left(\frac{m_3}{2}\right), Q_3 + \text{ceiling}\left(\frac{m_3}{2}\right), \text{if } d_3 \text{ is even}, \\ (Q_x - \text{ceiling}\left(\frac{m_3}{2}\right), Q_3 + \text{floor}\left(\frac{m_3}{2}\right), \text{if } d_3 \text{ is odd} \end{cases} \tag{10}$$

Example of extraction

Step 1

Stego-pixel block

P'_x=95	P'_1=103
P'_3=98	P'_2=106

Stego-quotient block

Q'_x=47	Q'_1=51
Q'_3=49	Q'_2=53

Stego-L block

L'_x=1	L'_1=1
L'_3=0	L'_2=0

Step 2

$L'_x = 1$, QVD extraction shall be applied. $R_1 = L'_1$=1, $R_2 = L'_2$=0, and $R_3 = L'_3$=0

$d_1 = \|Q'_x - Q'_1\| = 4$	$d_2 = \|Q'_x - Q'_2\| = 6$	$d_3 = \|Q'_x - Q'_3\| = 2$
d_1 belongs QR1, so LB_1=0	d_2 belongs QR1, so LB_2=0	d_3 belongs to QR1, so LB_3=0
$b_1 = d_1 - LB_1 = 4 = 100_2$	$b_2 = d_2 - LB_2 = 6 = 110_2$	$b_3 = d_3 - LB_3 = 2 = 010_2$
$D_8D_7D_6$=100_2	$D_5D_4D_3$=110_2	$D_2D_1R_4$=010_2

Step 3

$L'_x = 1$, so, the LSB extraction mentioned in step 3 of the extraction procedure is not applicable.

Step 4

$R'_1 = R_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$ $= 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 0$	$R'_2 = R_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$ $= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$
$R'_3 = R_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8$ $= 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 0$	$R'_4 = R_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8$ $= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$

Step 5

$EP = R'_4 \times 8 + R'_3 \times 4 + R'_2 \times 2 + R'_1$=0, so no error is found.

FIGURE 9. The example of data extraction and error correction.

it will be zero. This is the case of no error. If EP value is non-zero, then find the error bit referring to Table 2 and invert that bit. Note that this correction is valid if only 1 of the bits from the 8 bits $D_8D_7D_6D_5D_4D_3D_2D_1$ is erroneous.

Figure 9 depicts an example of data extraction and error correction step-by-step.

IV. RESULTS AND DISCUSSION

We have tested the proposed QVD+MHC scheme with more than 100 color images. Each pixel of a color image is 3 bytes. Figure 10 (a) depicts the 3 channels red, green, and blue. If the size of each channel is 512 × 512, then we can convert it to a 2-dimensional (2D) array of bytes by concatenating the channels, and we will get the size of the 2D array as 512 × 1536, as shown in Fig.10(b). The 2D array is raster scanned to form 2 × 2 disjoint blocks. Each element of such a block is considered as a byte. Each byte is treated as a grey image pixel, and all computations (data embedding, data extraction, PSNR calculation, QI calculation etc.) are performed accordingly.

The developed QVD+MHC scheme is executed in a computing system with i5 processor using MATLAB. The input images are gathered from SIPI database. Figure 11 lists original samples, and the respective SIs are listed in Fig.12. In each SI 8.4 lakhs bits of data is hidden.

The efficacy is measured through HC, bits per byte (BPB), time of embedding (EmT), time of extraction (ExT), PSNR,

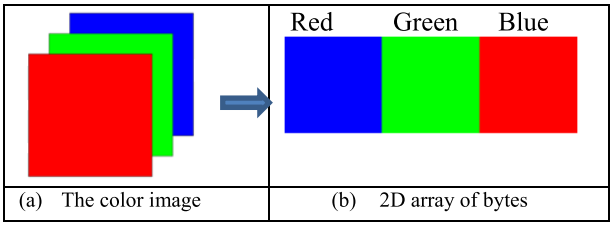


FIGURE 10. Converting a color image to a 2D array of bytes.

and quality index (QI). PSNR is an estimate of distortion in the SI. It is measured by (23), as shown at the bottom of page 10, wherein P_{ij} and Q_{ij} are the pixels of OI and the SI respectively.

HC is the magnitude of data in bits the image can conceal. The per byte HC is known as BPB. Furthermore, the likeliness between OI and SI is computed as QI in (24), as shown at the bottom of page 10.

Table 3 records the efficacy measures of the proposed technique. It can be noticed that the mean PSNR value over the 8 sample images is 36.76. Although PSNR value above 40 dB is always good achievement, within 30 to 40 dB is also acceptable. The QI is 0.9977, it implies a greater similarity between the OI and SI. The HC per byte is 3 bits and total HC in the image is 2359296 bits. Due to the inclusion of MHC and exclusive-or operation the embedding time has been larger, in average it is 34.02 seconds (Sec). The extraction time is

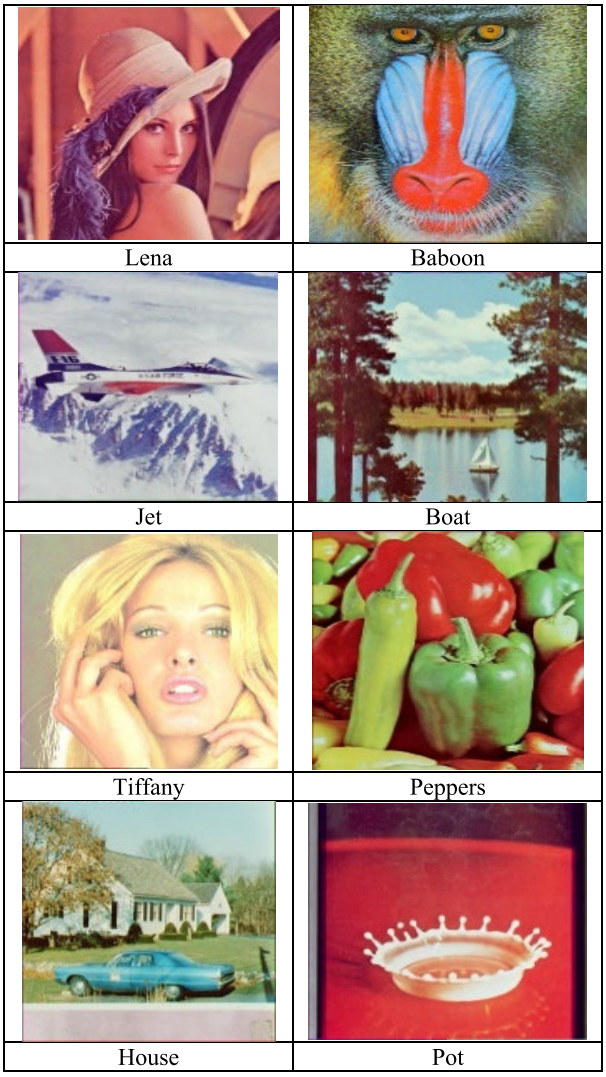


FIGURE 11. A set of OIs before hiding data in them.

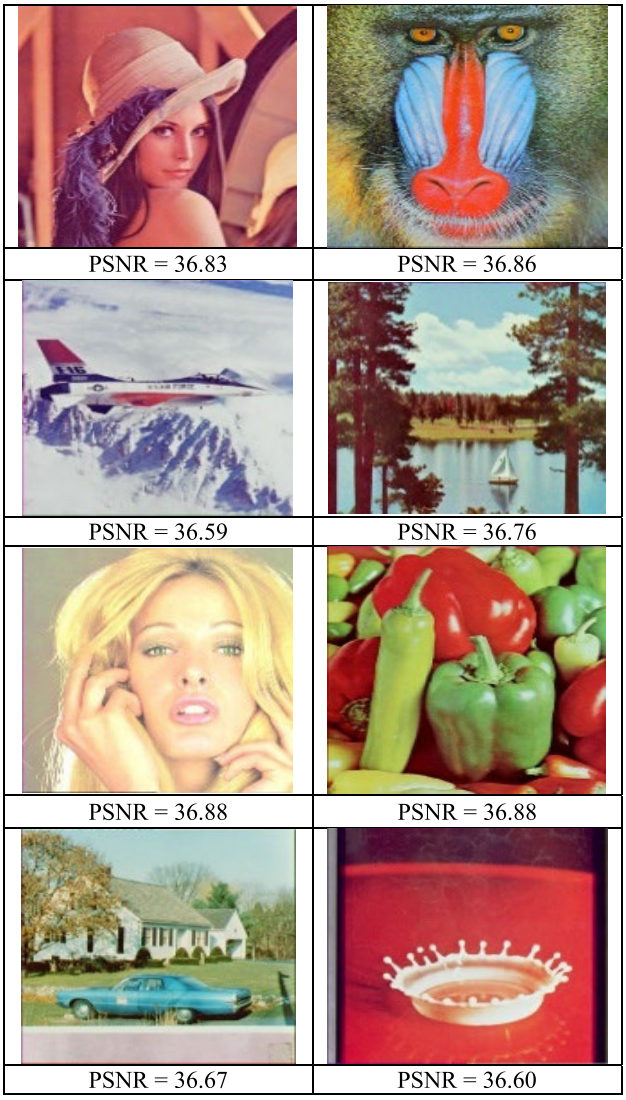


FIGURE 12. A set of SIs after hiding eight lakhs, and forty thousand bits of data in each of them.

TABLE 3. Performance of QVD+MHC scheme.

Images	PSNR (dB)	HC (bits)	QI	BPB	EmT (Sec)	ExT (Sec)
Lena	36.83	2359296	0.9981	3	31.94	8.22
Baboon	36.86	2359296	0.9979	3	36.02	8.31
Tiffany	36.88	2359296	0.9962	3	35.22	8.49
Peppers	36.88	2359296	0.9985	3	33.82	8.27
Jet	36.59	2359296	0.9964	3	34.02	8.25
Boat	36.76	2359296	0.9985	3	34.73	8.31
House	36.67	2359296	0.9977	3	33.63	8.24
Pot	36.60	2359296	0.9987	3	32.78	8.29
Average	36.76	2359296	0.9977	3	34.02	8.29

only 8.29 Sec. The overall performance is good. The focus is error detection and correction. Only 1-bit error can be detected and corrected accurately.

Table 4 represents a comparison study of this QVD+MHC scheme with some related existing techniques. The bar graph in Fig.13(a) distinguishes the bpb and PSNR values of the QVD+MHC scheme with existing schemes. The bar graph in Fig.13(b) distinguishes the QI value of the QVD+MHC scheme with existing schemes. It can be noticed here that the PSNR of the QVD+MHC scheme is higher than the 3 related existing schemes. PSNR of the QVD+MHC scheme is 36.76 dB, PSNR of Jung's scheme is 35.27dB, PSNR of Pradhan et al.'s scheme is 33.02 dB, and PSNR of Sonar & Swain's scheme is 35.15 dB. The QI value of the QVD+MHC scheme is higher than all the existing schemes. The QI value of QVD+MHC scheme is 0.9977, the QI value of Jung's scheme is 0.9967, the QI value of Pradhan et al.'s scheme is 0.9947 and the QI value of Sonar & Swain's scheme is 0.9966. The HC of the QVD+MHC scheme is 3 bits per byte, it is less than all the existing schemes. But the main objective of the QVD+MHC is to achieve error detection and correction, so HC is compromised.

TABLE 4. Comparisons of quality parameters.

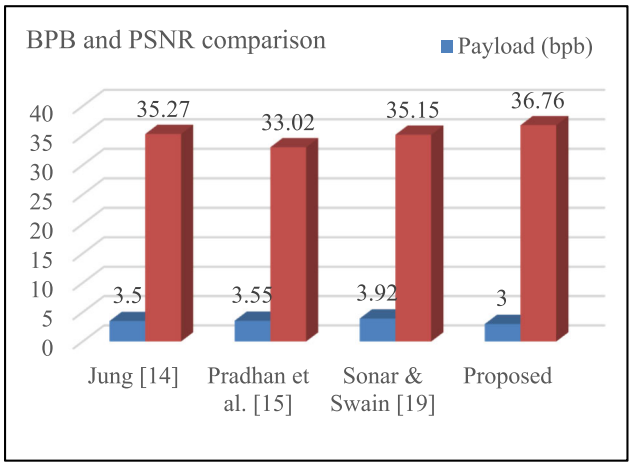
Techniques	Average PSNR	Average HC	Average QI	Average BPB
Jung's QVD [14]	35.27	2757637	0.9967	3.50
Pradhan et al.'s ASQVD [15]	33.02	2794301	0.9947	3.55
Sonar & Swain's QVD+PVC [19]	35.15	3086396	0.9966	3.92
Proposed QVD+MHC	36.76	2359296	0.9977	3.0

TABLE 5. Comparisons of error correction ability.

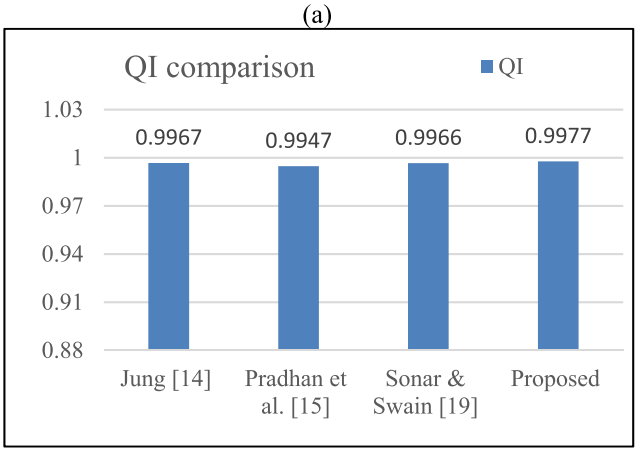
Techniques	Error detection	Error correction
Jung's QVD [14]	No	No
Pradhan et al.'s ASQVD [15]	No	No
Sonar & Swain's QVD+PVC [19]	No	No
Proposed QVD+MHC	Yes	Yes

Table 5 represents the error correction capabilities of the techniques. The existing schemes cannot perform error detection and correction, but this QVD+MHC scheme can perform 1-bit error detection and correction over every 8 bits of data extracted from a 4-pixel block. If there are more than 1 bit error, it cannot perform the error detection and correction.

Table 6 represents the comparison of imperceptibility with techniques having same or lower BPB. With a BPB of 3.13, the Khodaei and Faez's [9] technique gives slightly higher i.e., 38.57 dB PSNR, and 0.9984 QI. But it does not have error detection and correction abilities. The technique of Pradhan et al. [8] possesses slightly higher PSNR and QI by reducing the BPB. Furthermore, it does not have the ability for error detection and correction. The technique of Swain and Pradhan [21] possess lesser BPB, but slightly higher PSNR and QI. It possesses only error detection ability, does not possess error correction ability. The proposed scheme, and the schemes [8] and [21] uses the similar differencing mechanism referred from Wu and Tsai [2]. The PSNR of the proposed scheme is lower than that of [8] and [21] because the BPB is improved in proposed scheme. The scheme [9] uses addition and subtraction based differencing principle of Khodaei and Faez, so it possesses both higher PSNR and BPB as compared to that of the proposed scheme.



(a)



(b)

FIGURE 13. (a) Comparison of BPB, and (b) Comparison of QI.

TABLE 6. Comparisons of imperceptibility with techniques having same or lower BPB.

Techniques	PSNR (dB)	QI	BPB	Error detection, correction
Khodaei & Faez [9]	38.57	0.9984	3.13	Not available
Pradhan et al. [8]	39.78	0.9985	2.39	Not available
Swain & Pradhan [21]	39.74	0.9988	2.21	Only error detection
Proposed scheme	36.76	0.9977	3.00	Both are available

Table 7 records the PSNR values of the Lena and Baboon images with different HC. It can be noticed that the PSNR

$$PSNR = 10 \times \log_{10} \frac{m \times n \times 255 \times 255}{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(P_{ij} - Q_{ij}\right)^2} \tag{23}$$

$$QI = \frac{4 \times \bar{P} \times \bar{Q} \times \left\{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(P_{ij} - \bar{P}\right) \times \left(Q_{ij} - \bar{Q}\right)\right\}}{\left\{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(P_{ij} - \bar{P}\right)^2 + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(Q_{ij} - \bar{Q}\right)^2\right\} \times \left\{\left(\bar{P}\right)^2 + \left(\bar{Q}\right)^2\right\}} \tag{24}$$

TABLE 7. PSNR with increasing HC.

HC in bits	PSNR for Lena image	PSNR for Baboon image
7,000	57.21	57.57
14,000	54.22	54.69
1,40,000	44.55	44.62
2,80,000	41.58	41.68
4,20,000	39.84	39.94
5,60,000	38.59	38.70
7,00,000	37.62	37.74
8,40,000	36.83	36.86
9,80,000	36.16	36.18
11,20,000	35.57	35.62

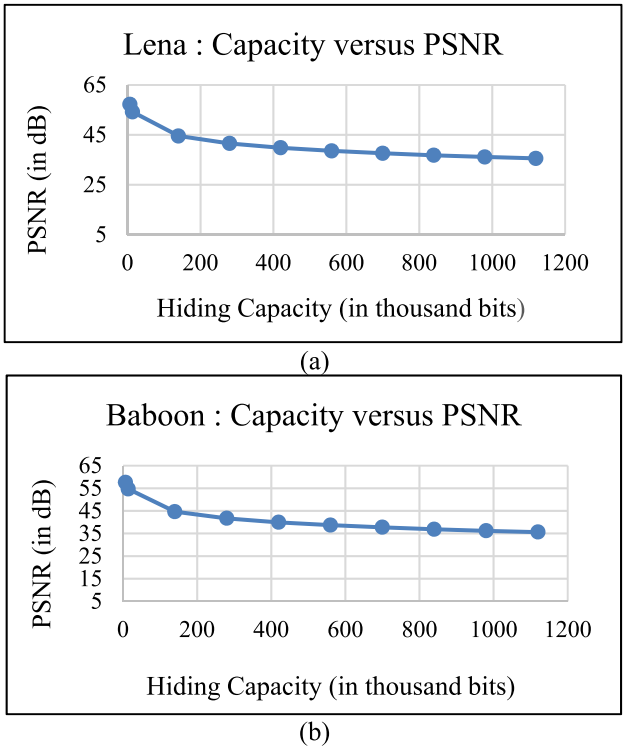


FIGURE 14. Capacity versus comparison PSNR, (a) for Lena image, and (b) for Baboon image.

goes on decreasing when the magnitude of concealed data keeps on increasing. The graphs in Figs. 14 (a) and (b) show capacity versus PSNR for the Lena and Baboon images respectively. It can be noticed that, although the PSNR goes on decreasing with the increase in hidden data, it will never fall below 35 dB because the curve approaches to be a horizontally parallel line.

RS analysis is conducted to assess the security of this QVD+MHC scheme. Four parameter values "R_m, R_{-m}, S_m and S_{-m}" are calculated [35]. If "R_{-m} - S_{-m} > R_m - S_m", then the RS analysis successfully detected the image as SI. This is because the divergence between R_{-m} and S_{-m} is greater than the divergence between R_m and S_m. If "$R_m \approx R_{-m} > S_m \approx S_{-m}$", then RS analysis could not detect the image as SI. "$R_m \approx R_{-m} > S_m \approx S_{-m}$" means that R_m and R_{-m} will be parallel and close to each other, S_m and S_{-m} will be parallel and close to each other. The line R_m and R_{-m} are on upper side as compared to the lines S_m and S_{-m} with regard to the Y-axis. Figure 15 (a) shows RS analysis for image Lena, and figure 15 (b) shows RS analysis for image Baboon. In figures 15 (a) and 15 (b), the x-axes stand for the percentage of HC, and the y-axis stands for the % of regular (R_m and R_{-m}) and singular (S_m and S_{-m}) groups. It can be noticed from these 2 figures that "$R_m \approx R_{-m} > S_m \approx S_{-m}$" is mostly satisfied. Thus, it is proved that RS analysis could not detect the SI.

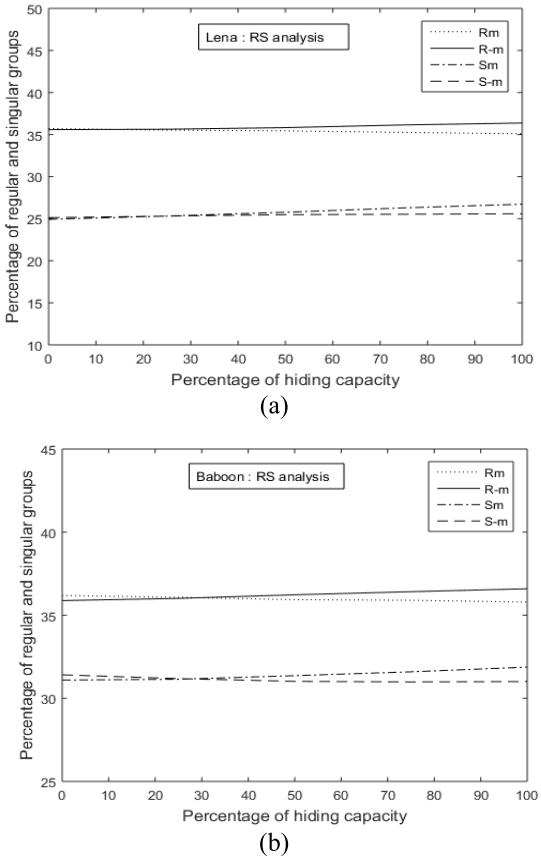


FIGURE 15. (a) RS analysis over Lena image, (b) RS analysis over Baboon image.

PDH analysis is performed to assess the security of this QVD+MHC scheme [35]. Figure 16(a) represents the PDH analysis of Lena image, and Fig.16(b) represents the PDH analysis of Baboon image. The SI considered here conceals hundreds of bytes of data. In these figures the dotted line curve stands for the pixel difference versus frequency of pixel difference for stego-image. The solid line curve stands for the pixel difference versus frequency of pixel difference for original image. As the dotted line curve is smooth in nature

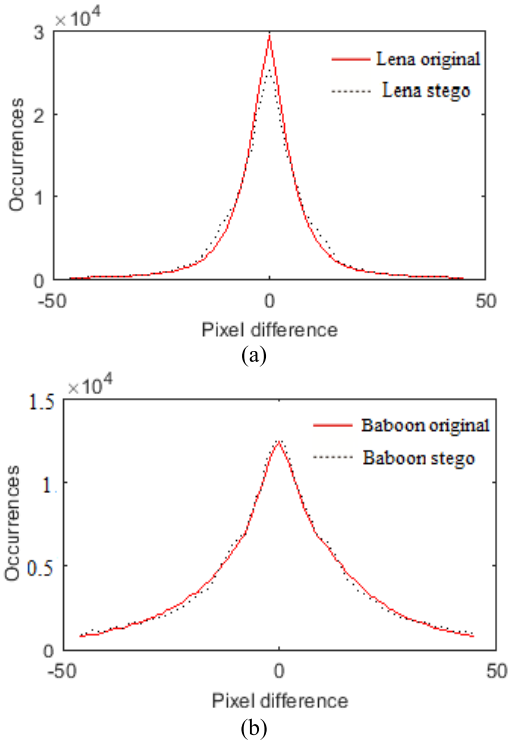


FIGURE 16. (a) PDH analysis over Lena image, (b) PDH analysis over Baboon image.

(no zig-zag shape is available), it implies that PDH analysis does not detect this steganography method.

V. CONCLUSION

The traditional image steganography techniques do not have the ability to detect errors and correct errors during the time of extraction at the receiver side. This article addresses this problem using MHC with the different bits of an image pixel. The image is divided into 2 × 2 dis-joint blocks. From these 4 pixels, 4 quotients and 4 LSBs are generated. Each quotient is decimal value of 7 MSBs of a pixel. For every block 8 data bits are camouflaged. From these 8 data bits, 4 redundant bits are computed using MHC. The 8 data bits and one redundant bit are camouflaged in the quotients by either QVD or bit substitution. If in quotients camouflaging is performed using QVD, then indicator bit is set to 1. Otherwise, if in the quotients camouflaging is performed using bit substitution, then indicator bit is set to 0. The 3 remaining redundant bits along with the indicator bit are stored in the LSBs. At the receiver's place data can be extracted and the error correction procedure can be plied to correct any errors. The experimental outcomes report that the PSNR value is 36.76 and it is greater than the existing techniques. The QI value is 0.9977 and it is also greater than the existing techniques. Furthermore, error detection and correction over the 8 bits of extracted data is possible in the proposed technique. But it is not possible in the existing techniques. The limitation of this work is that only one bit error can be detected and corrected, in future we shall make it multiple bit error detection and correction by changing the error detection and correction strategies.

REFERENCES

[1] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE MultimediaMag.*, vol. 8, no. 4, pp. 22–28, Apr. 2001, doi: 10.1109/93.959097.
[2] D. C. Wu and W. H. Tsai, "A steganograhic method for images by pixel value differencing," *Pattern Recognit. Lett.*, vol. 24, no. 9, pp. 1613–1626, 2003, doi: 10.1016/S0167-8655(02)00402-6.
[3] X. Zhang and S. Wang, "Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security," *Pattern Recognit. Lett.*, vol. 25, no. 3, pp. 331–339, Feb. 2004, doi: 10.1016/j.patrec.2003.10.014.
[4] Y.-P. Lee, J.-C. Lee, W.-K. Chen, K.-C. Chang, I.-J. Su, and C.-P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Inf. Sci.*, vol. 191, pp. 214–225, May 2012, doi: 10.1016/j.ins.2012.01.002.
[5] K. A. Darbkh, A. K. Al-Dhamari, and I. F. Jafar, "A new steganographic algorithm based on multi directional PVD and modified LSB," *Inf. Technol. Control*, vol. 46, no. 1, pp. 16–36, Apr. 2017, doi: 10.5755/j01.itc.46.1.15253.
[6] N. M. Ganguly, G. Paul, S. K. Saha, and D. Burman, "A PVD based high capacity steganography algorithm with embedding in non-sequential position," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 13449–13479, May 2020, doi: 10.1007/s11042-019-08178-9.
[7] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEE Proc.-Vis., Image Signal Process.*, vol. 152, no. 5, pp. 611–615, Oct. 2005, doi: 10.1049/ip-vis:20059022.
[8] A. Pradhan, K. Raja Sekhar, and G. Swain, "Digital image steganography based on seven way pixel value differencing," *Indian J. Sci. Technol.*, vol. 9, no. 37, pp. 1–11, Oct. 2016, doi: 10.17485/ijst/2016/v9i37/88557.
[9] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Process.*, vol. 6, no. 6, pp. 677–686, Aug. 2012, doi: 10.1049/iet-ipr.2011.0059.
[10] G. Swain, "Digital image steganography using eight-directional PVD against RS analysis and PDH analysis," *Adv. Multimedia*, vol. 2018, pp. 1–13, Sep. 2018, doi: 10.1155/2018/4847098.
[11] A. K. Shukla, A. Singh, B. Singh, and A. Kumar, "A secure and high-capacity data-hiding method using compression, encryption and optimized pixel value differencing," *IEEE Access*, vol. 6, pp. 51130–51139, 2018, doi: 10.1109/ACCESS.2018.2868192.
[12] M. A. Hameed, M. Hassaballah, S. Aly, and A. I. Awad, "An adaptive image steganography method based on histogram of oriented gradient and PVD-LSB techniques," *IEEE Access*, vol. 7, pp. 185189–185204, 2019, doi: 10.1109/ACCESS.2019.2960254.
[13] R. Kumar, D.-S. Kim, and K.-H. Jung, "Enhanced AMBTC based data hiding method using Hamming distance and pixel value differencing," *J. Inf. Secur. Appl.*, vol. 47, pp. 94–103, Aug. 2019, doi: 10.1016/j.jisa.2019.04.007.
[14] K.-H. Jung, "Data hiding scheme improving embedding capacity using mixed PVD and LSB on bit plane," *J. Real-Time Image Process.*, vol. 14, no. 1, pp. 127–136, Jan. 2018, doi: 10.1007/s11554-017-0719-y.
[15] A. Pradhan, K. R. Sekhar, and G. Swain, "Image steganography using add-sub based QVD and side match," in *Digital Media Steganography*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 81–97, doi: 10.1016/B978-0-12-819438-6.00013-X.
[16] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2995–3004, Apr. 2019, doi: 10.1007/s13369-018-3372-2.
[17] H.-H. Liu, Y.-C. Lin, and C.-M. Lee, "A digital data hiding scheme based on pixel-value differencing and side match method," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12157–12181, May 2019, doi: 10.1007/s11042-018-6766-y.
[18] S. Singh, "Adaptive PVD and LSB based high capacity data hiding scheme," *Multimedia Tools Appl.*, vol. 79, nos. 25–26, pp. 18815–18837, Jul. 2020, doi: 10.1007/s11042-020-08745-5.
[19] R. Sonar and G. Swain, "Steganography based on quotient value differencing and pixel value correlation," *CAAI Trans. Intell. Technol.*, vol. 2021, pp. 1–16, Jan. 2021, doi: 10.1049/cit2.12050.
[20] D. B. Khadse and G. Swain, "Data hiding using quotient value differencing and remainder value substitution avoiding incorrect extraction problem," *Sens. Imag.*, vol. 22, no. 1, pp. 1–21, Dec. 2021, doi: 10.1007/s11220-021-00360-4.

[21] G. Swain and A. Pradhan, "Image steganography using remainder replacement, adaptive QVD and QVC," *Wireless Pers. Commun.*, vol. 123, no. 1, pp. 273–293, Mar. 2022, doi: 10.1007/s11277-021-09131-6.
[22] C.-M. Wang, N.-I. Wu, C.-S. Tsai, and M.-S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *J. Syst. Softw.*, vol. 81, no. 1, pp. 150–158, Jan. 2008, doi: 10.1016/j.jss.2007.01.049.
[23] W. Zhao, Z. Jie, L. Xin, and W. Qiaoyan, "Data embedding based on pixel value differencing and modulus function using indeterminate equation," *J. China Universities Posts Telecommun.*, vol. 22, no. 1, pp. 95–100, 2015, doi: 10.1016/S1005-8885(15)60631-8.
[24] G. Swain, "A data hiding technique by mixing MFPVD and LSB substitution in a pixel," *Inf. Technol. Control*, vol. 47, no. 4, pp. 714–727, Dec. 2018, doi: 10.5755/j01.itc.47.4.19593.
[25] P. Maniriho and T. Ahmad, "Information hiding scheme for digital images using difference expansion and modulus function," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 31, no. 3, pp. 335–347, Jul. 2019, doi: 10.1016/j.jksuci.2018.01.011.
[26] Z. Li and Y. He, "Steganography with pixel-value differencing and modulus function based on PSO," *J. Inf. Secur. Appl.*, vol. 43, pp. 47–52, Dec. 2018, doi: 10.1016/j.jisa.2018.10.006.
[27] W. Tang, B. Li, W. Luo, and J. Huang, "Clustering steganographic modification directions for color components," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 197–201, Feb. 2016, doi: 10.1109/LSP.2015.2504583.
[28] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1905–1917, Sep. 2015, doi: 10.1109/TIFS.2015.2434600.
[29] M. Elhoseny, G. Ramírez-González, O. M. Abu-Elnasr, S. A. Shawkat, N. Arunkumar, and A. Farouk, "Secure medical data transmission model for IoT-based healthcare systems," *IEEE Access*, vol. 6, pp. 20596–20608, 2018, doi: 10.1109/ACCESS.2018.2817615.
[30] F. Li, K. Wu, X. Zhang, J. Yu, J. Lei, and M. Wen, "Robust batch steganography in social networks with non-uniform payload and data decomposition," *IEEE Access*, vol. 6, pp. 29912–29925, 2018, doi: 10.1109/ACCESS.2018.2841415.
[31] D. R. I. M. Setiadi, S. Rustad, P. N. Andono, and G. F. Shidik, "Digital image steganography survey and investigation (goal, assessment, method, development, and dataset)," *Signal Process.*, vol. 206, May 2023, Art. no. 108908, doi: 10.1016/j.sigpro.2022.108908.
[32] S. Rustad, D. R. I. M. Setiadi, A. Syukur, and P. N. Andono, "Inverted LSB image steganography using adaptive pattern to improve imperceptibility," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3559–3568, Jun. 2022, doi: 10.1016/j.jksuci.2020.12.017.
[33] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 897–911, Mar. 2022, doi: 10.1109/TDSC.2020.3004708.
[34] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 685–696, Mar. 2020, doi: 10.1109/TCSVT.2019.2896270.
[35] G. Swain, "Two new steganography techniques based on quotient value differencing with addition-subtraction logic and PVD with modulus function," *Optik*, vol. 180, pp. 807–823, Feb. 2019, doi: 10.1016/j.ijleo.2018.11.015.

S. N. V. J. DEVI KOSURU received the M.Sc. degree in mathematics from Dr. B. R. Ambedkar University, India, in 2005, the M.C.A. degree from Jawaharlal Nehru Technological University, Hyderabad, India, in 2008, and the M.Tech. degree in computer science and engineering (CSE) from Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India, in 2012. She is currently pursuing the Ph.D. degree with the Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India. She has ten years of teaching experience. Her research interests include data hiding and image tamper detection using watermarking.

ANITA PRADHAN received the B.Tech. degree in computer science and engineering (CSE) from the Biju Patnaik University of Technology (BPUT), Rourkela, Odisha, India, in 2010, the M.Tech. degree in CSE from Jawaharlal Nehru Technological University (JNTU), Kakinada, Andhra Pradesh, India, in 2012, and the Ph.D. degree in CSE from the Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, in 2020. She is currently an Assistant Professor with the Department of CSE, Koneru Lakshmaiah Education Foundation. She has more than eight years of teaching/research experience. She has authored/coauthored more than ten research articles. Her research interests include image steganography, image watermarking, and networks.

K. ABDUL BASITH received the B.Tech. degree from Jawaharlal Nehru Technological University, in 2003, and the M.Tech. degree in computer science and engineering (CSE) from VT University, in 2007. He is currently pursuing the Ph.D. degree with the Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India. He is with the Department of CSE, Marri Laxman Reddy Institute of Technology and Management, Hyderabad. He has more than 19 years of teaching experience. He has guided several projects in UG and PG levels. He has authored two text books. His research interests include wireless sensor networks and network security. He was a recipient of the Best Teacher Award from the Institute of Aeronautical Engineering.

RESHMA SONAR received the M.Tech. degree in information technology and the Ph.D. degree in computer science and engineering from the Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India, in 2008 and 2022, respectively. She is currently an Associate Professor with the Department of Artificial Intelligence and Machine Learning, ISBM College of Engineering, Pune, Maharashtra, India. She has more than 21 years of teaching experience and published more than 20 research articles in international journals and conferences. Her research interests include image steganography, cryptography, and information security.

GANDHARBA SWAIN received the M.C.A. degree from the VSS University of Technology, Burla, Odisha, India, in 1999, the M.Tech. degree in computer science and engineering from NIT Rourkela, in 2004, and the Ph.D. degree in computer science and engineering from SOA University (world QS ranked), Bhubaneswar, in 2014. He is currently a Professor and the Head of the Department of Artificial Intelligence and Data Science, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. He has more than 24 years of teaching/research experience. He has guided five Ph.D. students and H-index of 30, until July 2023. He has authored two text books, and authored more than 90 research articles. His research interests include data hiding, image watermarking, and block chain technology. He was a recipient of the Best Teacher Award from the GMR Institute of Technology and Koneru Lakshmaiah Education Foundation. He has also received the Distinguished Researcher Award from the Koneru Lakshmaiah Education Foundation. He is listed among top 2% of researchers of the world by Elsevier/Stanford University, in 2020, 2021, and 2022. He is a reviewer of many reputed journals published by IEEE, ACM, Elsevier, Springer, T&F, InderScience, Hindawi, and Wiley.