International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 09 | Sep 2023 www.irjet.net p-ISSN: 2395-0072

IMAGE STEGANOGRAPHY WITH CONTRAST ENHANCEMENT USING HISTOGRAM SHIFTING

Dr. T. Madhavi Kumari[1], T. Anvesh[2]

[1](ECE, JNTUH College of Engineering Hyderabad, India)
[2](ECE, JNTUH College of Engineering Hyderabad, India)

Abstract –

Reversible data hiding (RDH) is a method for concealing data in digital media that can be used to convey additional data invisibly. The original image may be restored without any loss of quality using the RDH technique, and the concealed messages can be precisely extracted. RDH is helpful for tagging digital image applications. Additionally, it is possible to precisely restore the original content prior to people downloading. A digital image is compressed in early RDH methods to free up extra space for tolerating more data. Some picture contents, such as the unimportant bitplanes, are compressed and joined with extra bits. We propose creating the transfer matrix during histogram shifting by maximizing the histogram's entropy. The designated image with the extra data has a higher contrast than the original image after embedding. On the recipient's end, the extra information can be precisely removed, and the original image can be restored without any loss in quality. The proposed method can achieve a better embedding payload when compared to current RDH-CE methodologies.

***Key Words*: RDH, Histogram Shifting, Contrast Enhancement, Data Embedding, Transfer Matrix**

1. INTRODUCTION

Reversible data hiding (RDH) is a technology that conceals data in a digital medium in a way that is unnoticeable and is used to carry more data. The original image can be restored without any loss while the concealed messages can be precisely extracted using the RDH technique. RDH is helpful in labelling applications for digital photographs. Numerous RDH-related works have been completed in the last 20 years, the majority of them focus on reducing the MSE distortion. RDH traditionally comes in three forms including the histogram shifting based RDH, the difference expansion based RDH, and the lossless compression based RDH.

A digital image is compressed in early RDH methods to free up extra space for tolerating more data. In, certain image contents, such as the inconsequential bitplanes, are compressed and concatenated with the extra bits. The original image can be restored by decompression on the recipient end after the hidden bits have been read directly from the image's end. It is simple and effective to deploy the LC-based RDH. But the LC based RDH's compromise between embedding rate and image distortion falls short. A DE-based RDH that expands the differences of two adjacent pixels to carry extra bits has been proposed to increase embedding performance. Numerous more DE techniques were put forth based on this concept in an effort to increase embedding capacity or lessen image distortion. The prediction error expansion (PEE) is a typical approach. The estimation mistakes are enlarged to hold more bits once each pixel has been estimated using numerous neighboring pixels. Histogram shifting, another well-liked RDH concept, achieves embedding by reversibly altering an image's histogram. Histogram shifting involves moving some of an image's histogram bins to make space for more empty bins. Once the empty bins are filled, the extra bits are then inserted into the image.

2. LITERATURE REVIEW

[1] Y.-Q. Shi, X. Li, X. Zhang, H.-T.Wu, and B. Ma: Reversible data hiding (RDH), also known as lossless or invertible data hiding, has progressively grown to be a very active study subject in the field of data hiding during the past two decades. This has been supported by a growing number of publications published recently in the field of RDH research on topics that are becoming more widely discussed.[2] J. Fridrich, M. Goljan, and R. Du: In this research, we offer two brand-new approaches to invertible watermarking-based digital picture authentication. The new techniques are invertible in the sense that, if the image is determined to be authentic, the distortion caused by authentication may be eliminated to retrieve the original image data, whereas almost all watermarking approaches create some small degree of non-invertible distortion in the image.[3] J. Fridrich, M. Goljan, and R. Du: Since the original image is invariably warped as a result of data embedding constitutes a common flaw of almost all current data embedding techniques. Due to quantization, bit-replacement, or truncation at the grayscales 0 and 255, this distortion is often impossible to totally eradicate[4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber: We introduce a novel lossless (reversible) data-embedding method that allows for the precise recovery of the original host signal upon extraction of the embedded data..[5] J. Tian**:** Recently, reversible data embedding has attracted a lot of attention.

The original digital content can be fully recovered because the process is reversible. We provide a brand-new approach to reversible data-embedding for digital photos. In order to maintain low distortion and a very high embedding capacity, we investigate redundancy in digital images. [7] D. M. Thodi and J. J. Rodriguez: Due to its capacity to embed data with no host information loss, reversible watermarking has emerged as a highly sought-after subset of fragile watermarking for sensitive digital imaging in application areas including military and medical. Upon confirming the validity of the received content, this reversibility allows for the recovery of the original host content. A fresh reversible watermarking algorithm is what we suggest.

3. PROPOSED METHOD

First, we create an intermediate image based on the image. Then, we create a transfer matrix based on the transfer matrix. The transfer matrix relates to the properties of the intermediate image. Next, we use the vacancy reservation algorithm to create an intermediate image by embedding rooms. Then, we use the side information and encrypted additional data to iteratively embed the intermediate image into the embedded image. Finally, we recover the image by extracting the data. The data extraction and image recovery process is the opposite of the process of embedding.

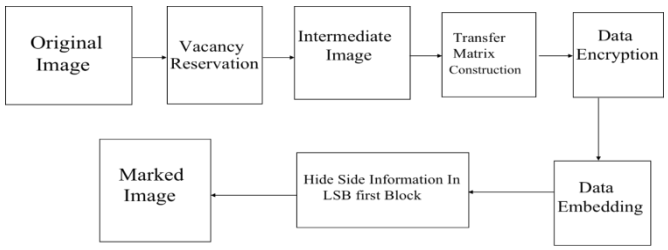

Figure 1 Block Diagram to Generate Marked Image

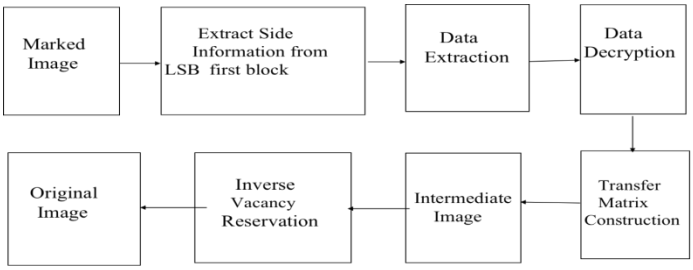

Figure 2 Block Diagram to recover Mark Original Image

A. Vacancy Reservation

In order to create the histogram h = {h0, h1, . . . , h255}, we first need a grayscale image of size mr x mc. We indicate that the original image has M original histogram vacancies, fulfilling hi = 0 (M might equal 0). We combine the least significant histogram bins (i.e., the bins with the fewest pixels) in h to create more histogram vacancies because using the original histogram vacancies from the original image can only promise a desirable embedding rate. ''Merging hi into hj'' refers to changing all pixels with the value i to value j in this context.

We select *N* least significant bins from h and denote them as h$LSBin$ = hL_1 , hL_2 , . . . , hL_N , such that $\forall hi \in h_{LSBin} \& \forall hj \in h - h_{LSBin}, \exists hi \le hj$. For each bin $h_{Li} \in h_{LSBin}$ ($1 \le i \le N$), We locate the closest non-empty bins (i.e., those that contain at least one pixel) from their left and right sides for each bin h_{Li} h_{LSBin} (1 i N). Let the left bin of h_{Li} be denoted by hl, and the right bin by hr. We combine h_{Li} with the hl and hr closest bin. Then, we change the values of all pixels with the value Li to hr or hl. Following these procedures, we get the post-merged picture I'. An suitable number of vacancies are set aside in its histogram, H0. We rescheduled after the reservation

The histogram vacancies are then assigned to h seq. A histogram vacancy must be filled in the neighbouring spot on either side in the final histogram if it is assigned with h^{seq} I. Here, the initial value of is a zero sequence.The information theory states that histogram shifting can include a total of g0log2 (1+wi) bits given the transfer possibility that a grayscale gi is reflected as (1+wi) grayscales in the marked image. To capture potential local entropy increase in the event that a histogram vacancy is assigned, we establish an entropy increment sequence e = e0, e1,..., eK.

Iterative steps are used to realise the vacant assignment. Assigning a histogram vacancy for each h0i in the first round will increase the overall histogram entropy by hi^{seq}. Thus e is given as e= hi^{seq} . We indicate hp,q ^{seq} as an element hi^{seq} .The maximum allowed in the r^{th} round is that. Additionally, we assign a free bin adjacent to the highest value in the h^{seq} by letting wp,1 =1, After every assignment round, the potential entropy rises.

$$e_{p,r} = h_{p,r}^{seq} log_2 \left(2 + \omega_{p,1}\right) - h_{p,r}^{seq} log_2 \left(1 + \omega_{p,1}\right)$$
$$= h_{p,r}^{seq} log_2 \left(\frac{2 + \omega_{p,1}}{1 + \omega_{p,1}}\right)$$

With each cycle, the potential embedding rate can be steadily elevated. Up until the total of equals N, the remainder of the vacancy assignment proceeds in the same manner. Then, in order to build the histogram for the enhanced picture Ie, we concurrently assign all of the histogram vacancies into the sequence.

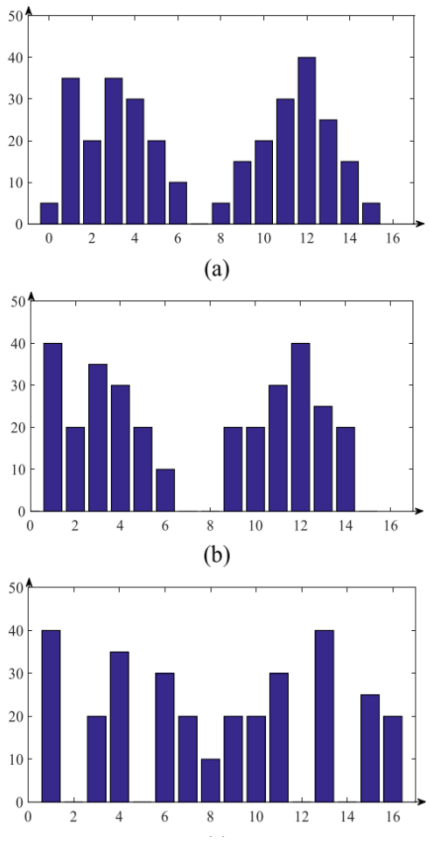

Figure 3 Example of vacancy reservation. (a) Original (b) after merging (c) after assigning the histogram vacancies.

gives an illustration of how reservations are made both the original image's histogram and. displays the merged image and histogram. The final image Ie and its histogram are shown.

B. Transfer Matrix Construction

The Transfer Matrix T indicate the process of transferring the histogram of the Enhanced Image to histogram of the Marked Image

$$\mathbf{T} = \begin{bmatrix} t_{0,0} & \cdots & t_{0,255} \\ \vdots & \ddots & \vdots \\ t_{255,0} & \cdots & t_{255,255} \end{bmatrix}$$

where $t_{i,j}$ Indicates the value of number of pixels changed from *i* to *j*. For a transfer matrix, the constraints for each *ti,j* is defined in

C. Data Encryption

The Hill Cipher is a traditional symmetric encryption method that uses matrix operations for both encryption and decryption and operates on blocks of plaintext. It bears Lester S. Hill's name and was created in 1929. Comparing the more secure Hill Cipher method to simpler substitution or transposition ciphers, the former is based on linear algebra and provides a better level of security. An overview of the Hill Cipher encryption method, its algorithm

i. Key Genration

Select a matrix, known as the encryption key matrix (K), with dimensions equal to the block size of plaintext. The key matrix should be invertible (non-singular) and have a modular multiplicative inverse, ensuring reversibility.

ii. Plaintext Encryption:

Break the plaintext into blocks of equal size and represent each block as a column vector (P). Multiply the key matrix (K) by each plaintext vector modulo a selected alphabet size (m). Convert the resulting matrix of ciphertext vectors back into letters to obtain the encrypted message.

Ciphertext vector (C) = (K * P) mod m

D. Data Embedding

After generating the transfer matrix T, we repeatedly insert additional data M into the image I_c. We separate the image I_c into blocks of the same size. By using the arithmetic decoding process, we embed a segment of additional data into each block B_k. We use the arithmetic decoding approach to include a piece of new data. The process of embedding is shown in the accompanying diagram, where the solid lines denote embedding and the dotted lines denote modifying the least significant bits. The data hiding and side information generating processes make up the embedding process.

Let block B_k contain s different types of pixel values.

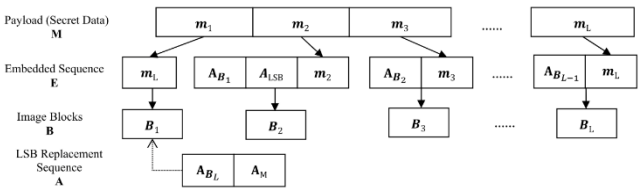

Figure 4 An illustration of the iterative data embedding process.

Let $b_{k,u}$ is a Vector of $b_{k,u}=[b_{k,u}, b_{k,u,\ldots} b_{k,u}]^T$ The possibility of changing a pixel value from i to j can be determined using the transfer matrix T.

$$p_i(j) = t_{i,j} \Big/ \sum_{k=0}^{255} t_{i,k}$$

We discover the vector $C_{k,u}$, which contains the pixel's initial value from $C1_{k,u}$. By encoding n($C_{k,u}$) integers using arithmetic using the possibilities, we produce a segment of side bits $A_{k,u}$.

$$A_{k,u}=Ari\_Enc(C_{k,u}, C1_{k,u,} P_{bk,U})$$

For pixel recovery in B_k, we then generate the side information A_{Bk}. Following the data embedding process for each block's values of pixels , We have the A_{bk} as side information for every block b_k. Then we embed the A_{bk} into the cover image We use A_{BL} and A_M to change the first block's least significant bits (LSB) in the final block. In reality, a single block's LSB length is always sufficient to hold the side information. Then, the picture should additionally contain the LSB of the first block's A_{LSB}.

When data concealing in the first block is complete, we move A_{LSB} ahead of M and change the LSBs with the binary sequence A = [A_{BL}, A_M]. You can directly extract ABL and AM from LSB as necessary side information to start the data extraction. By this process we generate the Marked Image.

E. Data Extraction

In this we extract the hidden data from the marked image Im. The data extraction process is done iteratively. During this process the recipient divide the image into same number of block as during the data embedding process L. from the first block's LSB exract Am and Abl. By using the side data Am we generate the transfer matrix by using the same steps mentioned in the transfer matrix process

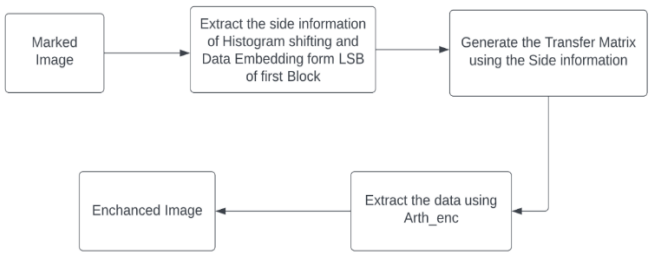

Figure 5 Block Diagram of Data Extraction

For extracting the message bits $M_{k,u}$ we need to use arthematic encoding of *Ck,u, Pk,u,. Here Pk,u, is the possibilities of the transfer matrix,*

$$\mathbf{P}'_{L,u} = \left[p_{L,u}(0), \ldots, p_{L,u}(255)\right]$$

$$\mathbf{m}_{k,u} = \text{Ari\_Enc}\left(\mathbf{c}_{k,u}, \bar{\mathbf{c}}_{k,u}, \mathbf{P}_{b_{k,u}}\right)$$

After Implementing the above equation for every block, we get each message bits stored in the block by combining those message blocks we get the total message. The message we get is the encrypted message we should decrypt the extracted message to get the final message.

F. Data Decryption

The decryption process in the Hill Cipher involves the inverse operation of the encryption process. To decrypt a ciphertext encrypted with the Hill Cipher, the recipient must possess the inverse of the encryption key matrix. The decryption process includes two main steps: key matrix inversion and ciphertext decryption. Key matrix inversion involves calculating the determinant of the key matrix, ensuring it is relatively prime to the modulus, finding the modular inverse of the determinant, adjoining the key matrix, and multiplying the adjoint matrix by the modular inverse of the determinant. Ciphertext decryption consists of breaking the ciphertext into blocks, converting each block into column vectors, multiplying the inverse key matrix by each column vector, taking the modulo 26 for each element, and converting the resulting vectors back into plaintext characters.

$P=K^{-1}*C(Mod26)$

Where p is the plain text, K is Key Matrix and C is the Cipher Text

G. Image Recovery

After extracting the message bits from the image, we get the contrast enhanced image without loss. To get the original image we perform inverse vacancy reservation. In this process we extract the side information from which is attached to additional message bit of the first block. With this side information we map the histogram bins which are non-empty to their original position.

4. RESULTS

This section includes all of the individual algorithm results, the overall results, and the statistical parameter results. Various photographs from the internet are used in our investigation, and the statistical factors are examined.

A. Input Images

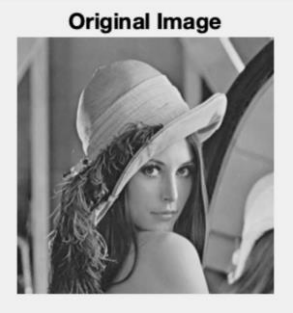

Figure 6(a) Lenna.

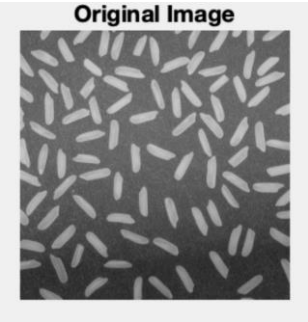

Figure 6(b) Grains

Figure 6(c) Bike

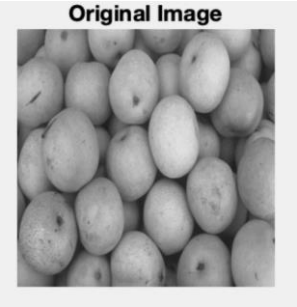

Figure 6(d) Pears

Figure 6(e) Coins

B. Images after Vacancy Reservation

In this the Images have higher contrast Original Images due to histogram Shifting

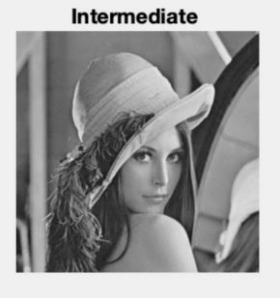

Figure 7(a) Intermediate Images of Lenna and Grains After Histogram Shifting

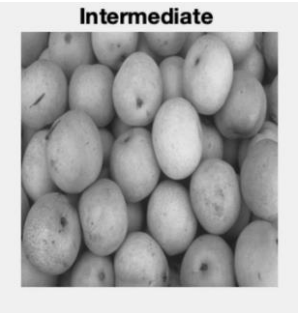

Figure 7(b) Intermediate Images of Bike and Pears After Histogram Shifting.

Figure 7(c) Intermediate Image of Coins. After Histogram Shifting

C. Data Encryption

In Data Encryption we use Hill Cipher Encryption method, which is a Substitution method, The original message is replaced with the cipher text. Original message which we want to encrypt can any message.

Here the Original message is “thehiddendata”

The Key Matrix used is 3X3 = {2,4,6,8,2,3,5,7,9}

```
Enter Input Message : thehiddendata
Enter The Key
element-2
element-4
element-6
element-8
element-2
element-3
element-5
element-7
element-9

key =

     2     4     6
     8     2     3
     5     7     9


determinant =

   42.0000

The Encrypted message is:
?{U%vsIlA]v[;P=
```

Figure 8(a) Hill Cipher Data Encryption

Here the Original message used is 100 Character long

The Key Matrix used is 3X3 = {3,10,10,20,9,17,9,4,17}

```
Enter Input Message : Reversibledatahiding(RDH)isatechr
Enter The Key
element-3
element-10
element-20
element-20
element-9
element-17
element-9
element-4
element-17

key =

     3    10    20
    20     9    17
     9     4    17


determinant =

  -1.6350e+03

The Encrypted message is:
L"Kop?cqdG0E*H#;P9\rqr;Quw'sXIYQh=!%_#DS/|GhQq4g>Y4N:T#
Your message is 24341 characters long.
This will require 24341 * 7 = 170387 pixels,
Your image has 231361 pixels so it will fit.
```

Figure 8(b) Hill Cipher Data Encryption

D. Marked Images (Images After Data Embedding)

After the Data encryption is done, we embed the cipher data into images Iteratively by using Arthematic decoding

Figure 9(a) Marked Images of Lenna and Grains.

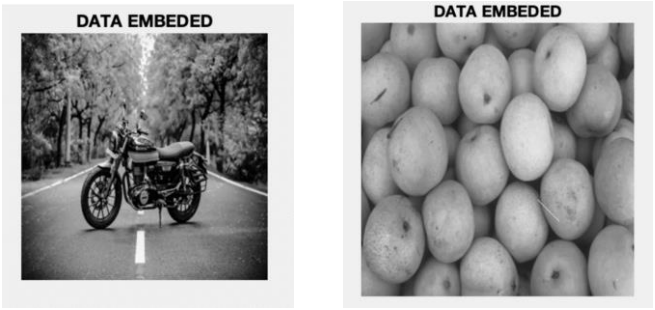

Figure 9(b) Marked Images of Bike and Pears

Figure 9(c) Marked Images of pears and Coins

E. Data Extraction

In this we extract the embedded data from the marked images iteratively by using arthematic encoding

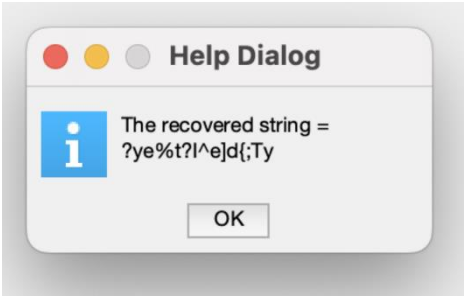

Figure 10(a) Image of Data Extraction

Figure 10(b) Image of Data Extraction

F.Data Decryption

To get the Original message we need covert the extracted cipher text to plain text to do this we use Hill cipher Decryption method

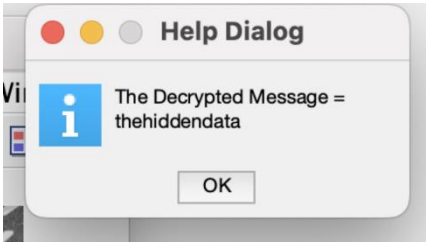

Figure 11(a) Image of Data Decryption using Hill Cipher

Figure 11(b) Image of Data Decryption using Hill Cipher

G. Image Recovery

As the data is extracted from the marked image, we get the intermediate image which is same as the intermediate images after the histogram shifting. Then we use the Inverse histogram shifting to map the merged histogram bins to back to their original position. By doing this we get the Original Image

Figure 12(a) Recovery of Original image of Leena and Grains

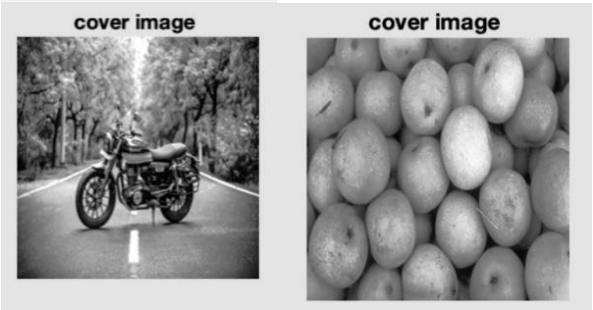

Figure 12(b) Recovery of Original image of Bike and Pears

Figure 12(c) Recovery of Original image of Pears and Coins

5. ANALYSIS

A. Peak Signal - to -noise Ratio

Peak Signal-to-Noise Ratio is a statistic for assessing how well a compressed or reconstructed image is doing. It calculates the difference between a signal's maximal power and the power of corrupting noise that degrades the accuracy of its representation. Two images are required to compute PSNR: the original image (also known as the reference image) and the compressed or reconstructed image whereas MSE is the Mean Squared Error between the two images, which is computed as: where MAXp is the maximum possible pixel value of the image (for instance, 255 for an 8-bit grayscale image).Here we get low PSNR values because during histogram shifting the pixels will be manipulated due to this the correlation between the pixel will decrease.

B. Structural Similarity Index (SSIM)

A technique for calculating how similar two images are is called the Structural Similarity Index (SSIM). Instead of focusing only on the pixel values, it compares the structural information between the two photos, which means it examines the patterns and textures in the images.

C. Relative Image Contrast (RCE)

An indicator of how accurately a display device can reproduce a range of brightness values is the relative contrast error. It is determined by dividing the actual contrast ratio by the measured contrast ratio and then expressing the result as a percentage. You must use a high-quality measurement tool to determine the display device's real contrast ratio in order to calculate the relative contrast inaccuracy. Calculate the difference between the two results after measuring the contrast ratio of the same display device with the same measurement tool.

D. Relative Entropy Error (REE)

REE Known also as Kullback-Leibler (KL) divergence, relative entropy error is a metric for comparing two probability distributions. It is frequently employed in statistics, machine learning, and information theory.

F. Relative Mean Brightness Error (RME)

The difference between the mean brightness values of the two images is quantified by the relative mean brightness error

Table 1- PSNR, SSIE, RCE Values between Input and Marked Images

IMAGE	PSNR	SSIM	RCE
Leena	24.6406	0.8974	0.4461
Grains	22.9999	0.8804	0.4337
Bike	22.1251	0.9150	0.4389
Pears	22.0546	0.8849	0.4314
Coins	25.7403	0.8912	0.4555

Table- REE and RMBE Values between Input and Marked Images

IMAGE	REE	RMBE
Leena	0.9756	0.9985
Grains	0.9938	0.9932
Bike	0.9933	0.9337
Pears	0.9883	0.9747
Coins	0.9372	0.9858

6. CONCLUSIONS

An innovative RDH-CE framework is put forward in this research. By combining the histogram's least significant bins, histogram vacancies are reserved in the baseline section. The transfer matrix for capacity expansion is proposed to be constructed using an iterative approach. The transfer probabilities for pixel alteration are computed using the transfer matrix. As a result, arithmetic coding can be used to incorporate the Encrypted extra information into the image . The experimental findings demonstrate that, as compared to current RDH-CE techniques, the suggested method can achieve a superior embedding payload while maintaining the visual quality of the marked image.

REFERENCES

[1] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, ‘‘Reversible data hiding: Advances in the past two decades,’’ *IEEE Access*, vol. 4, pp. 3210–3237,2016.

[2] J. Fridrich, M. Goljan, and R. Du, ‘‘Invertible authentication,’’ *Proc. SPIE,Secur. Watermarking Multimedia Contents III*, vol. 4314, pp. 197–208,Aug. 2001.

[3] J. Fridrich, M. Goljan, and R. Du, ‘‘Lossless data embedding—New paradigm in digital watermarking,’’ *EURASIP J. Adv. Signal Process.*, vol. 2002, no. 2, Dec. 2002, Art. no. 986842.

[4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, ‘‘Lossless generalized-LSB data embedding,’’ *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[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. Tian, ‘‘Wavelet-based reversible watermarking for authentication,’’ *Proc. SPIE, Secur. Watermarking Multimedia Contents IV*, vol. 4675, pp. 679–690, Apr. 2002.

[7] D. M. Thodi and J. J. Rodriguez, ‘‘Prediction-error based reversible water marking,’’ in *Proc. Int. Conf. Image Process.*, Oct. 2004, pp. 1549–1552.

[8] 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.

[9] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, ‘‘Reversible data hiding,’’ *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[10] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, ‘‘Reversible watermarking based on invariant image classification and dynamic histogram shifting,’’ *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 111–120, Jan. 2013.

[11] S.-K. Lee, Y.-H. Suh, and Y.-S. Ho, ‘‘Reversiblee image authentication based on watermarking,’’ in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1321–1324.

[12] X. Li, W. Zhang, X. Gui, and B. Yang, ‘‘Effificient reversible data hiding based on multiple histograms modifification,’’ *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 2016–2027, Sep. 2015.

[13] B. Ou, X. Li, Y. Zhao, and R. Ni, ‘‘Reversible data hiding using invari ant pixel-value-ordering and prediction-error expansion,’’ *Signal Process., Image Commun.*, vol. 29, no. 7, pp. 760–772, Aug. 2014.

[14] 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.

[15] Z. Qian, H. Zhou, X. Zhang, and W. Zhang, ‘‘Separable reversible data hiding in encrypted JPEG bitstreams,’’ *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 1055–1067, Nov./Dec. 2018.

[16] F. Huang, X. Qu, H. J. Kim, and J. Huang, ‘‘Reversible data hiding in JPEG images,’’ *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1610–1621, Sep. 2016.

[17] G. Xuan, Y. Q. Shi, Z. Ni, P. Chai, X. Cui, and X. Tong, ‘‘Reversible datahiding for JPEG images based on histogram pairs,’’ in *Proc. Int. Conf. Image Anal. Recognit.*, in Lecture Notes in Computer Science, vol. 4633. 2007, pp. 715–727.

[18] Z. Qian, H. Xu, X. Luo, and X. Zhang, ‘‘New framework of reversible datahiding in encrypted JPEG bitstreams,’’ *IEEE Trans. Circuits Syst. VideoTechnol.*, vol. 29, no. 2, pp. 351–362, Feb. 2019.