Hindawi
Security and Communication Networks
Volume 2023, Article ID 2068813, 14 pages
https://doi.org/10.1155/2023/2068813

Research Article

High Payload Lossless Steganography Using Image Interpolation

Raju Pratap Sharma,[1] Aruna Malik,[1] Samayveer Singh,[1] Saurabh Agarwal,[2] and Rajeev Kumar[3]

[1]*Department of CSE, Dr. BR Ambedkar NIT Jalandhar, Jalandhar, Punjab, India*
[2]*Department of Software Convergence, Andong National University, Andong, Republic of Korea*
[3]*Department of CSE, DTU, Delhi, India*

Correspondence should be addressed to Aruna Malik; malika@nitj.ac.in

Received 10 March 2023; Revised 8 November 2023; Accepted 13 November 2023; Published 28 November 2023

Academic Editor: Goutham Reddy Alavalapati

Interpolation-based reversible data hiding (IRDH) is a method for embedding data in a reversible manner into digital images. It is based on the polynomial interpolation to hide the secret data bit stream in the coefficients of the interpolation polynomial. In this paper, an IRDH method that combines a modified neighbor mean interpolation is presented with a new method for reversibly embedding secret data in interpolated pixels. This method uses log base 2 of the decimal value of the interpolated pixel to replace a corresponding number of bits with the bits of the secret data, allowing for a higher bit stream to be embedded with better image quality. The results from experiments show that the introduced method can implant on average, 1300000 bits of secret data with an average PSNR of 28.5 dB, which is a significant improvement over existing and related works.

1. Introduction

After the development of the Internet, information security has become one of the foremost apprehensions which results in a wide range of research areas in this field. Information in digital form is more vulnerable to attack such as copying, modification, and/or deletion. For this, some methods have been discussed to provide security for sharing data over the Internet. One of them is data hiding that allows the embedding of secret information in a given medium that may be a multimedia file such as image, audio, video, or text. Data hiding makes sharing of confidential information over the public network an easy task. Thus, it finds applications in the military, medical imaging, and diplomacy. Data hiding usually alters the values of pixels of cover images to conceal sensitive information so as to prevent its unauthorized peeking or damage [1, 2]. In 1983, Simmons was arguably the first to propose the idea of data hiding [3, 4]. Afterward, several data hiding methods (DHMs) have been presented by numerous researchers to minimize the tradeoff between embedding capacity and image quality. Among them, the least significant bit (LSB) substitution method has been the most popular and simple [5]. However, the LSB substitution method generally warps the original images permanently that carry private data. However, it is forbidden to compromise the original cover in forensic, military surveillance and communication systems, digital watermarking, and medical image processing and analysis applications. This drawback was addressed by the development of a technique that, after the abstraction of secret information, can reestablish the original cover image deprived of falsification. The procedure is called reversible data hiding (RDH) [6, 7].

RDH is a method for embedding secret data into cover media content in a way that allows for the retrieval of both the secret data and the original content without any falsification. The concept of RDH was first introduced in the 1990s and has since then been an active area of research [8–13]. RDH is based on the idea that it is likely to embed additional data into a digital multimedia file without modifying the original data. The main challenge in implementing RDH is to ensure that the unique content is recuperated without any degradation or loss while allowing the

hidden data to be extracted and recovered. There are several algorithms and techniques used in RDH and the effectiveness of RDH techniques is typically evaluated based on three factors: imperceptibility, robustness, and hiding capacity. Imperceptibility refers to the ability of the technique to hide the data without affecting the quality of the original content, robustness refers to the ability of the technique to withstand attacks such as compression and cropping, and capacity refers to the amount of data that can be hidden within the media content. Overall, RDH is a valuable tool for adding security and protection to digital multimedia content. It allows for the embedding of important information without compromising the quality of the original content.

In the last decades, several RDH approaches have been developed which include difference expansion [14], histogram shifting [15], prediction error expansion [16, 17], and interpolation [9, 18, 19]. Among them, interpolation-based RDH techniques are quite popular in medical imaging as these allow scaling of images while carrying the private data. The goal of image interpolation is to obtain the most accurate estimation of a pixel's intensity based on the values of its nearby pixels. Thus, it helps in creating high-resolution images from low-resolution ones. In this work, a new RDH-based method on modified neighbor mean interpolation is proposed which aims to embed a high quantity of secret information while ensuring better quality of the image. The contributions of the paper are presented as follows:

(i) The proposed work introduces interpolation-based reversible data hiding (IRDH) as a novel method for reversible data embedding in digital images. It employs polynomial interpolation to hide a secret data bit stream within the coefficients of the interpolation polynomial.

(ii) The paper presents a modified neighbor mean interpolation technique, which is integrated into the IRDH method. This technique enhances the accuracy and visual quality of the interpolated pixels, leading to improved image reconstruction.

(iii) This work proposes a new method for reversibly embedding secret data in interpolated pixels. Instead of directly replacing bits, the method utilizes the logarithm (base 2) of the decimal value of the interpolated pixel. This logarithmic transformation allows for the selective substitution of a corresponding number of bits with the bits of the secret data, enabling higher data embedding capacity without compromising image quality.

(iv) To evaluate the performance of the proposed method, the results demonstrate that, on average, the method can embed 1,300,000 bits of secret data with an average peak signal-to-noise ratio (PSNR) of 28.5 dB. This performance surpasses existing and related works, indicating a significant improvement in data embedding capacity while maintaining high-quality images.

In summary, the work contributes an IRDH method that combines a modified neighbor mean interpolation technique with a logarithmic-based reversible data embedding approach. This approach enhances image reconstruction accuracy and allows for a higher data embedding capacity. The experimental results validate the effectiveness of the method and establish its superiority over existing techniques in terms of both data embedding capacity and image quality.

The remaining paper is prepared as follows. In Section 2, we have deliberated some traditional approaches of various interpolation methods and some RDH methods based on them. In Section 3, we proposed our interpolation approach following the information embedding and extraction process. In Section 4, we analyzed our result and compared it with some traditional techniques. Finally, the study is concluded in Section 5.

2. Related Works

The related work is distributed into two segments. In the first segment, some popular and existing image interpolation methods such as bilinear interpolation, nearest neighbor interpolation, bicubic interpolation, and neighbor mean interpolation are discussed. In the next subsection, some of the RDH methods based on image interpolation are momentarily studied.

2.1. Review of Existing Image Interpolation Methods. Image interpolation is the scaling/resizing of an image from a one-pixel grid to another. More specifically, image interpolation increases or decreases the whole pixels in the image which helps in zooming in for finer details. Some of the popular image interpolation methods are briefly discussed below.

(i) Nearest neighbor interpolation (NNI) is a modest interpolation method that chooses the closest value opinion to the user. The NNI for expanding an image involves two steps: (1) Pixel values are first assigned to newly created places once they have been constructed and (2) the freshly produced pixel value is given the closest neighbor pixel value.

(ii) Neighbor mean interpolation (NMI) was proposed by Jung and Yoo [1]. The NMI method calculates a mean to find the missing pixel value. It takes very less time in processing and finds the interpolated values.

(iii) Bilinear interpolation considers the two previously identified pixel values that are nearest and surround the unknown pixel or the neighborhood of those pixels. The final interpolated value is determined using a weighted average of four pixels, which provides significantly smoother images. The bilinear interpolation is computationally more complex and hence takes more time compared to NNI but provides a better image.

(iv) Bicubic interpolation considers a total of sixteen pixels for estimating the interpolated pixels. It basically goes one step further than bilinear by taking

into account the closest four-by-four neighborhood of previously known pixels. The method gives the surrounding pixels a higher weighting because they are located at varied distances from the unknown pixel. Bicubic produces images that are noticeably sharper than those produced by earlier techniques, and it might be the optimum tradeoff between processing speed and output quality.

2.1.1. Existing Interpolation-Based RDH Methods. In 2009, Jung and Yoo introduced the RDH method using neighbor mean interpolation (NMI) [1]. The method involves reducing the image size by half, then expanding it back to its original size using NNI to create a cover medium. Secret information is then hidden into the interpolated image by dividing it into 2×2 block of pixel and embedding the information into three of the four pixels in each block, except for the first pixel. However, the pixels are recomputed before extracting the information from the original image. In 2011, Jan et al. identified the weakness of Jung and Yoo's method by proposing a new data hiding method [20]. However, the weakness of Jung and Yoo's method is that the pixel values are recomputed completely before extracting the information from the original image. In 2011, Jan et al. proposed a new method that addresses this weakness by generating a new difference value for each interpolated pixel in the 2×2 pixel block, and using this value to determine the number of embeddable bits in a pixel. This method improves the embedding capacity while enhancing visual quality and eliminates the need to recalculate the interpolated pixel values. In 2012, Hu and Li [6] further extended the work of Jung and Yoo by using a different interpolation method and introducing a new data hiding method that determines the number of embeddable bits by computing the variance between the maximum values of the four nearest pixels [21]. In 2013, Chang et al. presented a new interpolation-based RDH technique that uses the ENMI interpolation method and embeds data in two layers. In 2015, Hu and Li extended the work of Lee and Huang [21] by expanding the difference between adjacent pixels to raise the embedding capacity. All of these methods aim to achieve a balance between good image quality and high data embedding capacity. This work proposes a new method that combines a modified neighbor mean interpolation with a new data embedding approach to further improve performance.

Brahma et al. introduce a pixel selection technique that incorporates the local variance of both the color channels and the grayscale image. This approach aims to enhance the quality of the selected pixels by prioritizing those with lower magnitude prediction errors, thereby reducing distortion. Additionally, to improve the embedding capacity of the RDH technique, the authors suggest double embedding in one of the color channels. This is achieved by further pixel selection among the already selected pixels. Fan et al. introduce a novel RDH method for interpolated images using modulo operation and prediction error expansion [22]. It effectively reduces image distortion, achieves high embedding capacity, and maintains a high PSNR. The method outperforms similar approaches in terms of image quality and demonstrates resistance against histogram analysis. Malik et al. introduce a novel interpolation method and a new reversible data hiding scheme for upscaling images and embedding secret data [23]. The proposed scheme considers human visual system characteristics to maintain visual quality while embedding a large amount of data. Experimental results demonstrate high performance compared to existing interpolation-based data hiding methods.

Xiong et al. [24] discuss an adaptive interpolation-based reversible data hiding (RDH) algorithm that improves visual quality and embedding capacity. By sorting interpolated pixels and employing coding techniques, it outperforms traditional RDH algorithms and demonstrates resistance to steganalysis attacks.

Huang et al. present a block-based adaptive RDH method for high fidelity [25]. It creates an interpolated image using reference pixels and embeds secret data into interpolated pixels with low noise levels. The method incorporates the center folding strategy for improved visual quality and ensures reversibility without overflow/underflow problems or additional overhead messages. Mandal et al. propose a high-capacity reversible and secure data hiding technique in images, utilizing interpolation and difference expansion techniques [7]. The method focuses on maximizing the difference between neighboring pixels while maintaining image quality. Hassan and Gutub discuss a scheme that enlarges the original image using ENMI interpolation and embeds data into interpolated pixels using a novel method based on pixel intensity and maximized difference values [26]. It covers all steps of image generation, data embedding, extraction, and recovery for fair comparison. Tripathi and Prakash discuss a reversible data hiding method based on an interpolation scheme and histogram shifting which considers a blockchain-based system to encrypt the images [27]. Bai et al. discuss an efficient scheme based on the differences between interpolation algorithms. By encoding secret data using distinct interpolated values, the receiver can retrieve the data through matching the sorted interpolation values [28]. The papers [29–31] discuss the digital image steganography survey and investigation regarding the goal, assessment, method, development, and dataset.

3. Proposed Method

A new interpolation-based reversible data hiding method (IRDH) is introduced in this section. The proposed work is organized in three parts. In Section 3.1, a new image interpolation, i.e., modified neighbor mean interpolation (MNMI) is proposed, then elaborating the procedure of data embedding in Section 3.2 followed by the illustrative example of the information embedding process. In Section 3.3, we describe the data extraction process followed by an illustrative example to showcase the same. Figure 1 shows the flowchart of the proposed method which explains the step-by-step complete process.

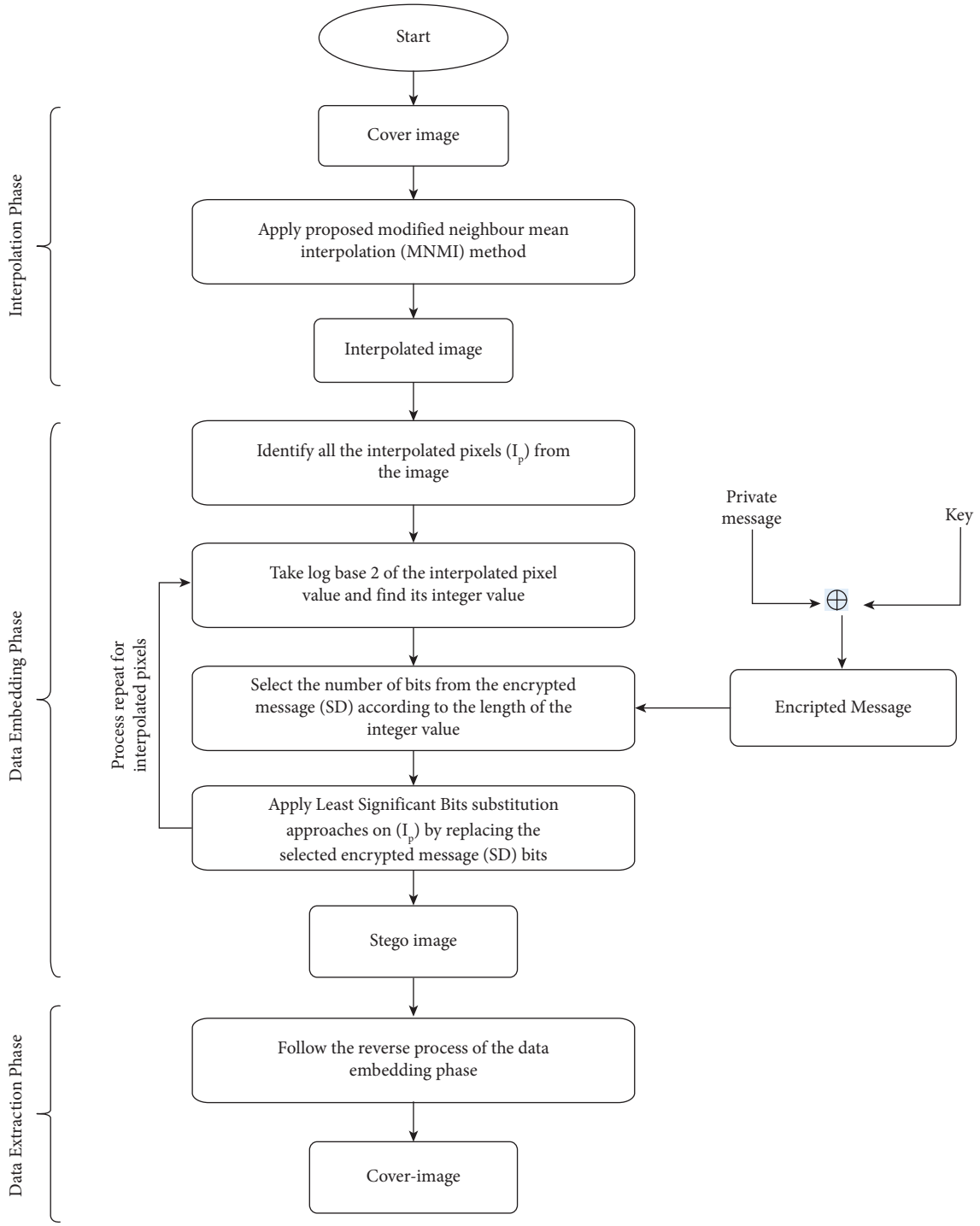


Figure 1: Flowchart of the proposed method.

3.1. Proposed Modified Neighbor Mean Interpolation (MNMI) Method. The proposed MNMI method is highly encouraged by the work of Chang et al. [4] and Malik et al. [32]. The working of the modified neighbor mean interpolation (MNMI) method is defined using the following Algorithm 1.

3.1.1. An Illustrative Example of the Proposed Image Interpolation Method. In this section, an example of image interpolation is described in detail by considering a 3×3 block from an original image as $O(x, y)$, where x and y are the dimensions of X and Y coordinate of image's pixel. The complete process of the converting original image into the interpolated image is shown in Figure 2. The pixels of the sub-block of the original image (7, 12, 95, 8, 6, 82, 40, 43, 35), whose locations are as follows: $O(0, 0), O(0, 1), O(0, 2), O(1, 0), O(1, 1), O(1, 2), O(2, 0), O(2, 1), O(2, 2)$. After applying the proposed algorithm of image interpolation, the 3×3 sub-block is converted to a 5×5 sub-block called

Input: Input Original image O of size $N \times N$
Output: Cover image C
for $x = 0$ to $N - 1$ do
  for $y = 0$ to $N - 1$ do

$$C(x,y) = \begin{cases} O(x,y) \text{ if } x \,\%\, 2 == 0 \text{ and } y \,\%\, 2 == 0, \\ O(x,y-1) \; if \; y{=}N-1, \\ O(x-1,y) \text{if } x{=}N-1, \\ [O(x,y-1) + O(x,y+1)]/2 \; if \; x \,\%\, 2 == 0 \text{ and } y \,\%\, 2 == 1, \\ [O(x-1,y) + O(x+1,y)]/2 \; if \; x \,\%\, 2 == 1 \text{ and } y \,\%\, 2 == 0, \\ [O(x-1,y) + O(x,y-1) + O(x+1,y) + O(x,y+1)]/4 \text{otherwise}, \end{cases}$$

  end for
end for

Algorithm 1: Modified neighbor mean interpolation (MNMI).

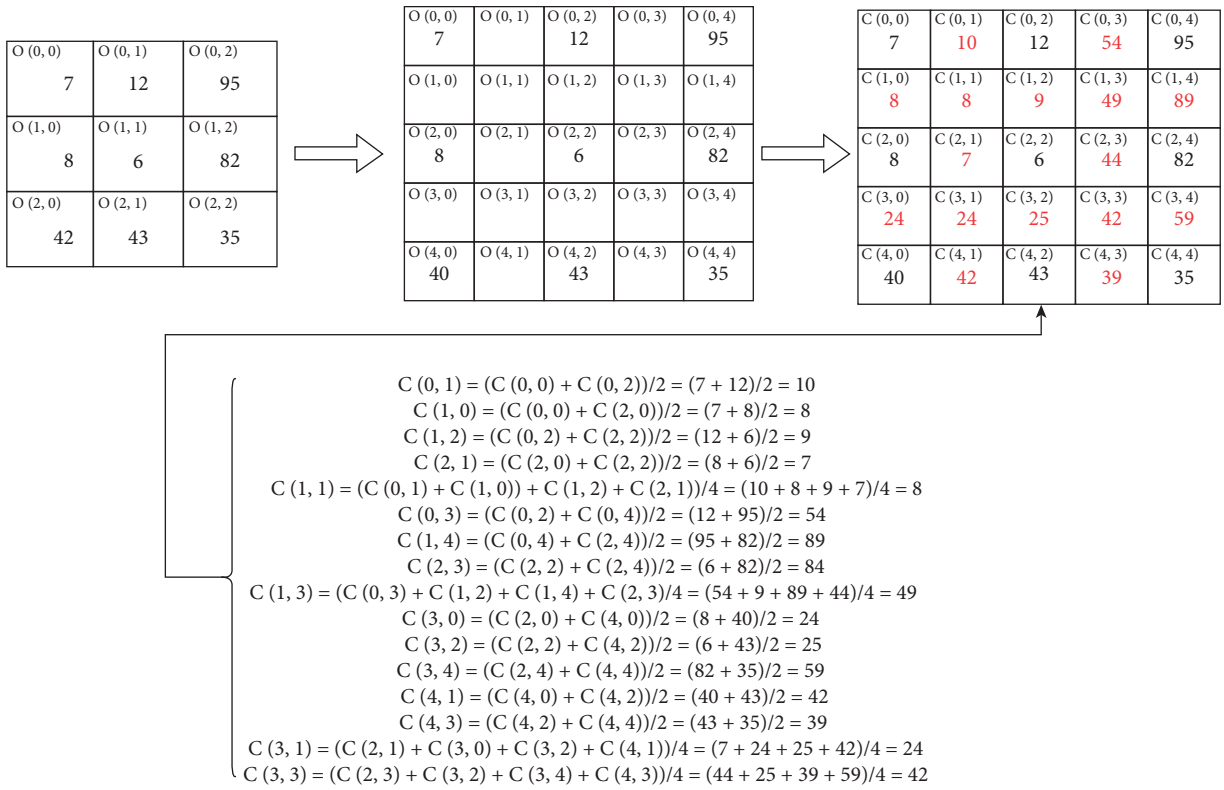


Figure 2: Process of converting the original image into an interpolated image.

a cover image, i.e., one space is created adjacent to all original pixels. With the help of the proposed algorithm, we find the missing value of pixels, and then, we call the new image as $C(x, y)$ cover image, such as $C(0, 1) = (C(0, 0) + C(0, 2))/2 = (7 + 12)/2 = 10$, $C(1, 0) = (C(0, 0) + C(2, 0))/2 = (7 + 8)/2 = 8$, $C(1, 2) = (C(0, 2) + C(2, 2))/2 = (12 + 6)/2 = 9$, $C(2, 1) = (C(2, 0) + C(2, 2))/2 = (8 + 6)/2 = 7$, $C(1, 1) = (C(0, 1) + C(1, 0) + C(1, 2) + C(2, 1))/4 = (10 + 8 + 9 + 7)/4 = 8$, $C(0, 3) = (C(0, 2) + C(0, 4))/2 = (12 + 95)/2 = 54$, $C(1, 4) = (C(0, 4) + C(2, 4))/2 = (95 + 82)/2 = 89$, $C(2, 3) = (C(2, 2) + C(2, 4))/2 = (6 + 82)/2 = 84$, $C(1, 3) = (C(0, 3) + C(1, 2) + C(1, 4) + C(2, 3))/4 = (54 + 9 + 89 + 44)/4 = 49$, $C(3, 0) = (C(2, 0) + C(4, 0))/2 = (8 + 40)/2 = 24$, $C(3, 2) = (C(2, 2) + C(4, 2))/2 = (6 + 43)/2 = 25$, $C(3, 4) = (C(2, 4) + C(4, 4))/2 = (82 + 35)/2 = 59$, $C(4, 1) = (C(4, 0) + C(4, 2))/2 = (40 + 43)/2 = 42$, $C(4, 3) = (C(4, 2) + C(4, 4))/2 = (43 + 35)/2 = 39$, $C(3, 1) = (C(2, 1) + C(3, 0) + C(3, 2) + C(4, 1))/4 = (7 + 24 + 25 + 42)/4 = 24$, and $C(3, 3) = (C(2, 3) + C(3, 2) + C(3, 4) + C(4, 3))/4 = (44 + 25 + 39 + 59)/4 = 42$. In this way, all missing pixels are calculated and the interpolated image is obtained.

3.2. Data Embedding Algorithm. In this section, the complete information embedding algorithm is discussed where the cover media/image is considered the input and the stego image will be the output.

Step 1: Choose a private message that is required to generate the secret data (SD).

Step 2: Choose a private key of eight bits and XOR the bit stream of the private key with every byte of the

private message. As a result, we obtain an encrypted secret message.

Step 3: Identify all interpolated pixels (I_P) of the resultant cover image (also called interpolated image) to embed the encrypted private message (SD).

Step 4: Take the log base 2 of the decimal value of the interpolated pixel and whatever value we obtain, just replace that much of bits of the binary value of the interpolated pixel by applying the least significant bits substitution approaches from the encrypted private message (SD).

Step 5: Repeat steps (1) to (4) for all the resultant's interpolated pixels for the current interpolated image block. Furthermore, the same process will be repeated for the complete cover image to obtain a stego image.

3.2.1. An Illustration of the Proposed Method with an Example. The comprehensive illustration of the process of information embedding in the proposed method is given in Figure 3. Let us consider the cover image block of the pixels as 7, 10, 12, 54, 95, 8, 8, 9, 49, 89, 8, 7, 6, 44, 82, 24, 24, 25, 42, 59, 40, 42, 43, 39, and 35. The encrypted secret message as " 10010101010011001101010 11 0101001110100011011110010 01011010010001000" is prepared by applying the XOR operation with every byte of the secret message to obtain the encrypted message in order to provide one extra layer of security. After that, we identify the interpolated pixel from the cover image, i.e., 10, 54, 8, 8, 9, 49, 89, 7, 44, 24, 24, 25, 42, 59, 42, and 40. Here, the first interpolation pixel (I_P) is 10 which is in binary as 00001010. Then, we calculate the log base 2 value of 10 as ($\log_2$ 10), i.e., 3.3. We consider the integer value of $\log_2$ value, i.e., 3. So, we can hide 3 bits from the secret data (100) to the interpolated pixel value (00001010) using the LSB method; the updated value will be 00001100 which is equal to 12 in decimal value. The next interpolated pixel (I_P) value is 54 which is in binary as 00110110. Moreover, the log base 2 integer value of 54 ($\log_2$ 54) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00110101 which is equal to 53 in decimal. The next interpolated pixel (I_P) value is 8 which is in binary as 00001000. And the log base 2 integer value of 8 ($\log_2$ 8) is 3. So, in this case, 3 bits of data can be hidden. After replacing 3 bits of interpolated pixel (I_P) using LSB, we will obtain 00001010 which is equal to 10 in decimal. The next interpolated pixel (I_P) value is 8 which is in binary as 00001000. And the log base 2 integer value of 8 ($\log_2$ 8) is 3. So, in this case, 3 bits of data can be hidden. After replacing 3 bits of interpolated pixel (I_P) using LSB, we will obtain 00001011 which is equal to 11 in decimal. The next interpolated pixel (I_P) value is 9 which is in binary as 00001001. And the log base 2 integer value of 9 ($\log_2$ 9) is 3. So, in this case, 3 bits of data can be hidden. After replacing 3 bits of interpolated pixel (I_P) using LSB, we will obtain 00001001 which is equal to 9 in decimal. The next interpolated pixel (I_P) value is 49 which is in binary as 00110001. And the log base 2 integer value of 49 ($\log_2$ 49) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00110101 which is equal to 53 in decimal. The next interpolated pixel (I_P) value is 89 which is in binary as 01011001. And the log base 2 integer value of 89 ($\log_2$ 89) is 6. So, in this case, 6 bits of data can be hidden. After replacing 6 bits of interpolated pixel (I_P) using LSB, we will obtain 01011010 which is equal to 90 in decimal. The next interpolated pixel (I_P) value is 7 which is in binary as 00000111. And the log base 2 integer value of 7 ($\log_2$ 7) is 2. So, in this case, 2 bits of data can be hidden. After replacing 2 bits of interpolated pixel (I_P) using LSB, we will obtain 00000110 which is equal to 6 in decimal. The next interpolated pixel (I_P) value is 44 which is in binary as 00101100. And the log base 2 integer value of 44 ($\log_2$ 44) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00101110 which is equal to 46 in decimal. The next interpolated pixel (I_P) value is 24 which is in binary as 00011000. And the log base 2 integer value of 24 ($\log_2$ 24) is 4. So, in this case, 4 bits of data can be hidden. After replacing 4 bits of interpolated pixel (I_P) using LSB, we will obtain 00011000 which is equal to 24 in decimal. The next interpolated pixel (I_P) value is 24 which is in binary as 00011000. And the log base 2 integer value of 24 ($\log_2$ 24) is 4. So, in this case, 4 bits of data can be hidden. After replacing 4 bits of interpolated pixel (I_P) using LSB, we will obtain 00011101 which is equal to 29 in decimal. The next interpolated pixel (I_P) value is 25 which is in binary as 00011001. And the log base 2 integer value of 25 ($\log_2$ 24) is 4. So, in this case, 4 bits of data can be hidden. After replacing 4 bits of interpolated pixel (I_P) using LSB, we will obtain 00011110 which is equal to 30 in decimal. The next interpolated pixel (I_P) value is 42 which is in binary as 00101010. And the log base 2 integer value of 42 ($\log_2$ 42) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00101001 which is equal to 41 in decimal. The next interpolated pixel (I_P) value is 59 which is in binary as 00111011. And the log base 2 integer value of 59 ($\log_2$ 59) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00101101 which is equal to 45 in decimal. The next interpolated pixel (I_P) value is 42 which is in binary as 00101010. And the log base 2 integer value of 42 ($\log_2$ 42) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00100100 which is equal to 36 in decimal. The next interpolated pixel (I_P) value is 40 which is in binary as 00101000. And the log base 2 integer value of 40 ($\log_2$ 40) is 5. So, in this case, 5 bits of data can be hidden. After replacing 5 bits of interpolated pixel (I_P) using LSB, we will obtain 00101000 which is equal to 40 in decimal. Therefore, this process continues until all the pixels are recovered of the cover image.

3.3. Data Extraction Process. The extraction approach is applied at the recipient side, i.e., the reverse procedure of information embedding in order to recover the private data and actual image. The algorithm's steps are given as follows:

Step 1: Identify the interpolated pixels (I_P) in the stego image in which we have hidden the secret embedding.

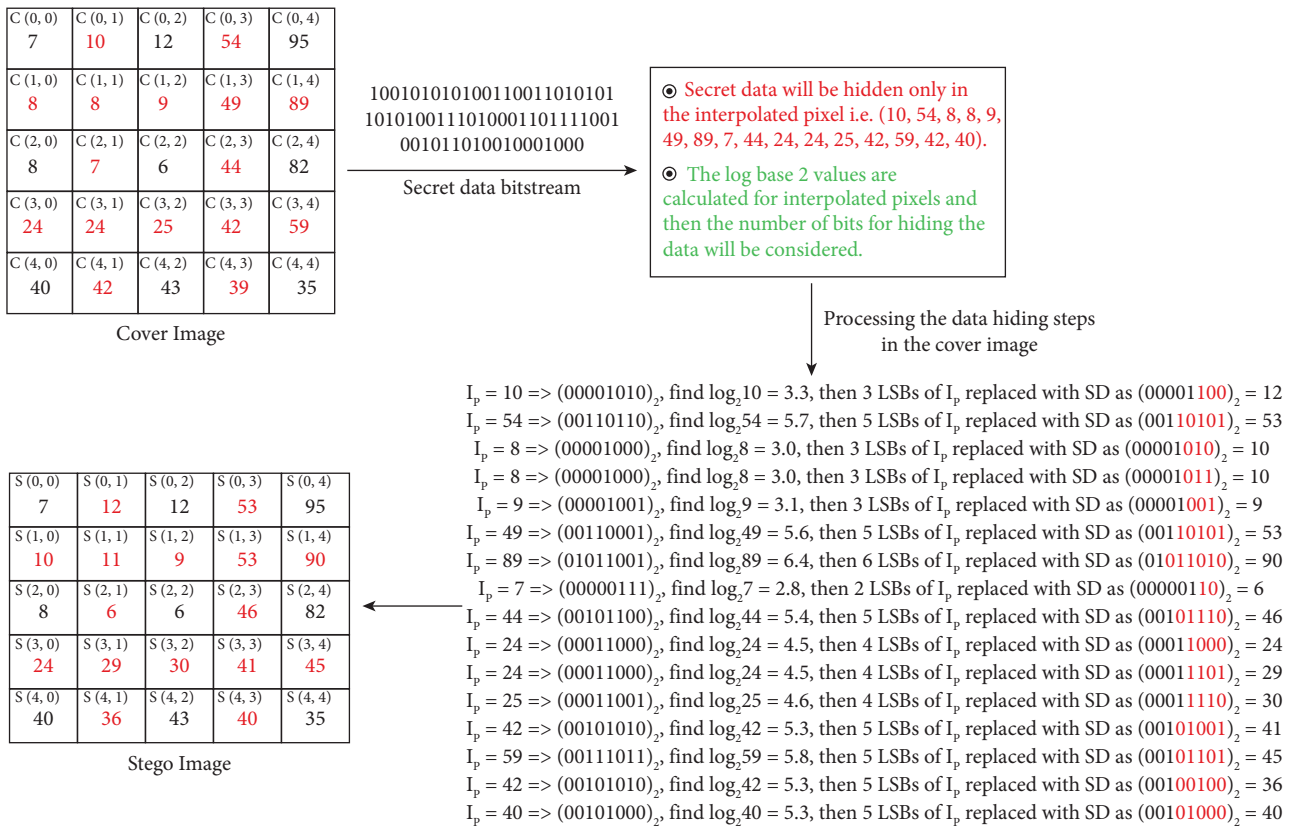


Figure 3: An illustration of the embedding process of the proposed work.

Step 2: Take log base 2 of identified interpolated pixel (I_P), whatever log value is obtained, extract that much LSB from the pixel, and then add these bits in the main secrete data bitstream.

Step 3: Repeat step 2, until all identified stego image pixels are extracted.

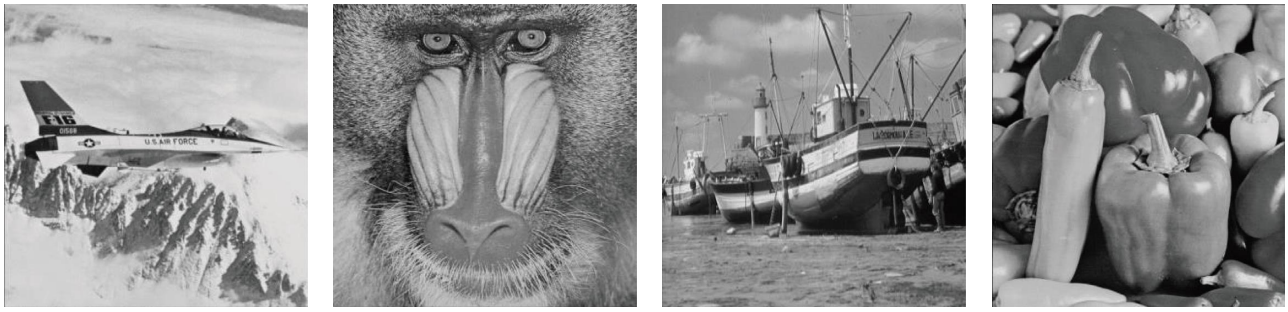
Step 4: Using the private key, XOR the achieved private data in bytes to obtain the original private data bitstream.

Step 5: To obtain the actual cover image, finally remove the interpolated pixels.

3.3.1. An Illustration of the Data Extraction. In this section, an illustration of the data extraction is discussed in detail. The preceding embedding phase produces a stego image at the recipient side and extraction of data is done in order to obtain the secret information and recover the original image. At first, we aimed to obtain the secret information from the stego image, so we have to identify the interpolated pixels in the stego image which are as follows: (12, 53, 10, 11, 9, 53, 90, 6, 46, 24, 29, 30, 41, 45, 36, 40). So, the first interpolated pixel is 12 whose binary value is 00001100. Now, we take log base 2 value of 12, we will obtain 3, so we extract 3 LSB and add to the main data bit stream and we will store that in a separate location. Again, the next interpolation is 53 whose binary value is 00110111. And the log base 2 integer value of 53 is 5. So, we extract 5 LSB of 53 and augment to the main secret information bit stream. The next interpolated stego pixel is 10 whose binary value is 00001010. We take log base 2 of 10 and we obtain 3. So, we extract 3 LSB and augment to the secret information bit stream. The next interpolated stego pixel is 11 whose binary value is 00001011 and the log base 2 value is 3. So, we extract 3 LSB and augment to the main secret information bit stream. Hence, by repeating the same process, we extract all the interpolated stego pixels. At last, we will obtain the complete data bit stream as "1001010101001100110101011010100111 0100011011111001 0010110100100010 00." After performing XOR operation on the bit stream and the private key, we will obtain the original secret message.

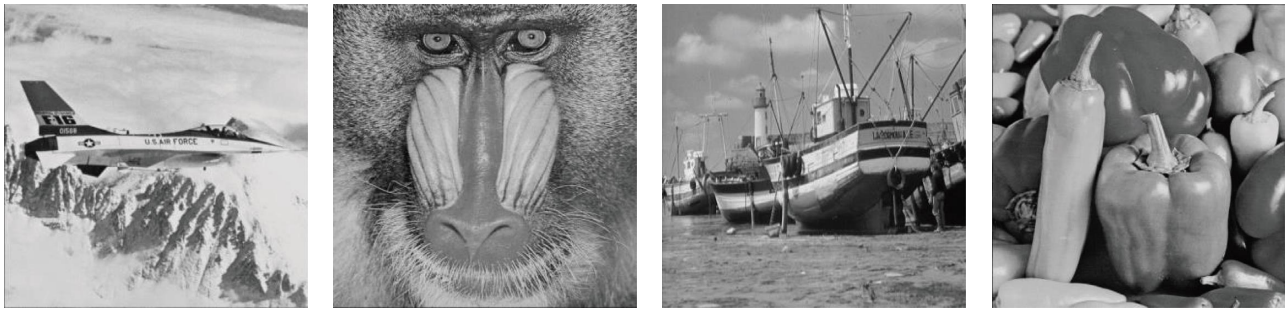
4. Result Analysis and Discussion

In this section, the experimental results of the proposed method are thoroughly deliberated. The performance of the proposed method is compared to some of the most widely used IRDH methods, providing insights into its effectiveness. To perform the experiments, four standard grayscale test images of size 512×512 pixels, as shown in Figure 4, were used. The proposed method was executed using Python on a system with 12th Gen Intel Core i7-1255U 1.7 GHz processor, 16 GB RAM, and x64-based architecture. Two key metrics are used to analyze the performance, including visual image quality measured using PSNR and the embedding capacity measured in bits. The stego images are shown in Figure 5.



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

Figure 4: Standard original image. (a) Airplane. (b) Baboon. (c) Boat. (d) Peppers.



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

Figure 5: Stego image. (a) Airplane. (b) Baboon. (c) Boat. (d) Peppers.

First of all, the comparative performance analysis of the proposed modified neighbor mean interpolation (MNMI) against some of the popular interpolation methods, which include NNI, bilinear, and NMI, is discussed. For this, the results are illustrated in Figure 6. The provided experimental results show that the proposed MNMI achieves PSNR greater than 30 for all the images except Baboon. It can also be experimentally observed from Figure 6 that the existing methods are not even close in terms of PSNR with the proposed MNMI method. More specifically, the PSNR of NNI method for Lena, Baboon, Airplane, and Peppers images is 22.56 db, 16.91 db, 21.03 db, and 22.22 db, respectively. Similarly, the PSNR of Bilinear Interpolation for Lena, Baboon, Airplane, and Peppers images is 26.24 db, 19.59 db, 24.24 db, and 25.47 db, respectively. Similarly, the PSNR of NMI for Lena, Baboon, Airplane, and Peppers images is 26.82 db, 20.09 db, 24.66 db, and 25.99 db, respectively. As far as the proposed MNMI is concerned, the PSNR for Lena, Baboon, Airplane, and Peppers images is 34.18 db, 29.78 db, 34.40 db, and 34.10 db, respectively, which is significantly higher than the aforementioned existing interpolation methods. Thus, it can be clearly stated that our interpolation technique is superior to the existing aforementioned interpolation methods.

Now, the presentation of the proposed IRDH method is compared against the existing IRDH methods which include Jung and Yoo [1], Zhang et al. [33], Lee & Huang [21], Tripathi and Prakash [27], Bai et al. [28], and the proposed method. Jung & Yoo's method [1] embeds secret data bits based on the image region's smoothness and complexity. So, the embedding capacity varies as per the cover image characteristics. The amount of data that Jung & Yoo's method [1] embeds varies between 1,77,830 and 4,28,240 bits for the given set of test images. As far as Chang et al. [9] embedding capacity is concerned, 1,84,280 to 3,82,841 bits could be hidden. Similarly, Zhang et al. [20] can conceal data ranging from 5,31,550 to 6,14,981 bits. As far as Lee & Huang's [7] embedding capacity is concerned, 3,42,520 to 6,40,938 bits could be hidden. The embedding capacity of Bai et al. [27] is 585225 bits for all the images. It is constant for all images. Similarly, Tripathi and Prakash [28] can conceal data ranging from 146249 to 250160 bits. The embedding capacity of the proposed method varies from 12,22,582 to 13,37,339 bits for the test images. The average embedding capacity achieved by Jung and Yoo's [1] is 2,54,983 bits with a PSNR of 27.79 dB. While the average embedding capacity achieved by Zhang et al. [33], Lee & Huang [21], Tripathi et al. [27], Bai et al. [28] and proposed approaches are 2,50,100, 5,59,328, 4,39,277, 5,85,225, 2,01,273, and 12,77,368 bits with 27.445 dB, 26.09 dB, 26.61 dB, 35.51 dB, and 28.91 dB PSNR, respectively, as shown in Figure 7.

The performance of the proposed RDH method is exhaustively analyzed and presented in Table 1. The results clearly indicate that the PSNR achieved by the proposed method is higher than all other IRDH methods, regardless of the test image used. In addition, Figure 6 provides a visual representation of the maximum embedding capacity achieved by the proposed method. These results affirm the

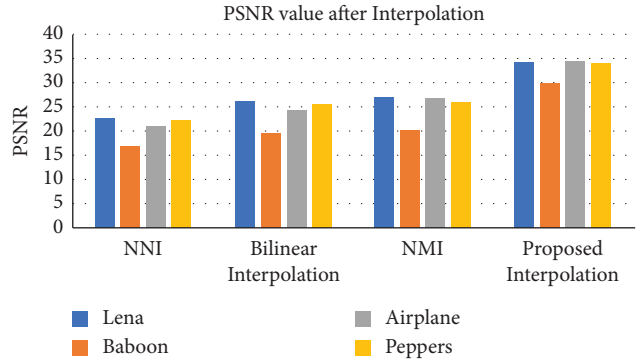


Figure 6: Comparison of PSNR value after interpolation.

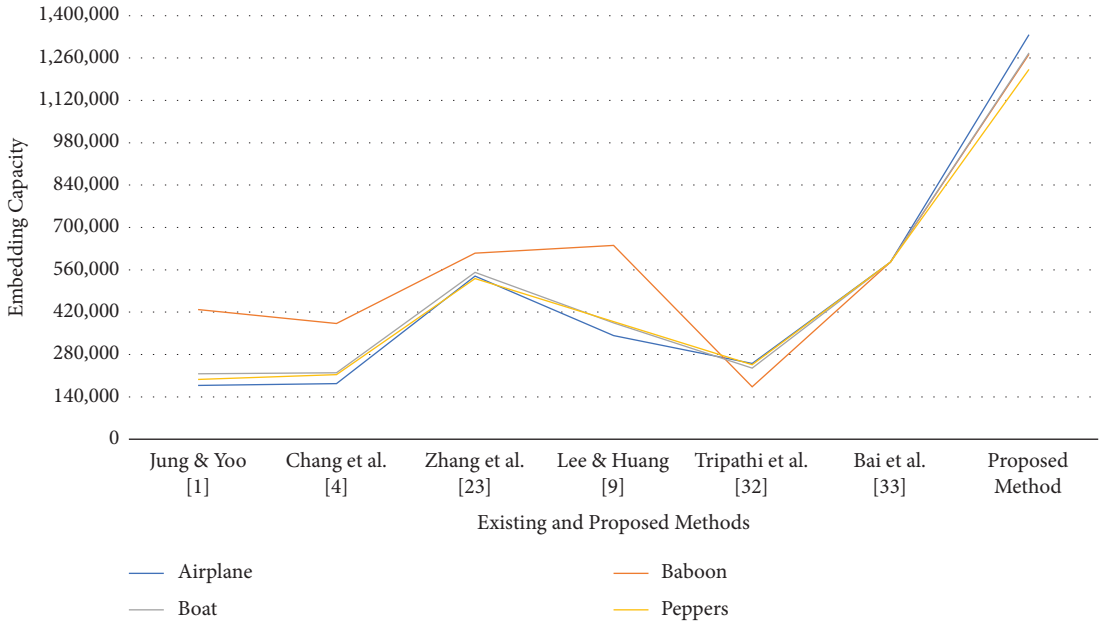


Figure 7: Comparison of data hiding capacity.

Table 1: Comparison of PSNR of the existing IRDH methods with the proposed method.

Test image	Jung and Yoo [1]	Zhang et al. [33]	Lee and Huang [21]	Tripathi and Prakash [27]	Bai et al. [28]	Proposed method
Airplane	28.61	28.71	27.38	28.39	35.12	28.98
Baboon	23.93	22.14	21.22	22.01	31.80	28.46
Boat	28.73	28.85	27.41	27.57	37.89	28.95
Peppers	29.91	30.08	28.35	28.49	37.24	29.28

Table 2: Comparison of embedding capacity in terms of bits per pixel of the existing IRDH methods with the proposed method.

Test image	Jung and Yoo [1]	Zhang et al. [33]	Lee and Huang [21]	Tripathi and Prakash [27]	Bai et al. [28]	Proposed method
Airplane	0.67836	2.05533	1.30661	0.95428	2.23245	5.10154
Baboon	1.63360	2.34596	2.44498	0.66094	2.23245	4.85472
Boat	0.82495	2.10565	1.46739	0.89806	2.23245	4.87104
Peppers	0.75380	2.02770	1.48384	0.55789	2.23245	4.66378

superiority of the proposed method over existing techniques. It is important to note that Table 1 provides a comprehensive evaluation of the proposed RDH method, highlighting its ability to deliver higher PSNR values compared to other IRDH methods. This information is further supported by the results illustrated in Figure 6, which demonstrate the

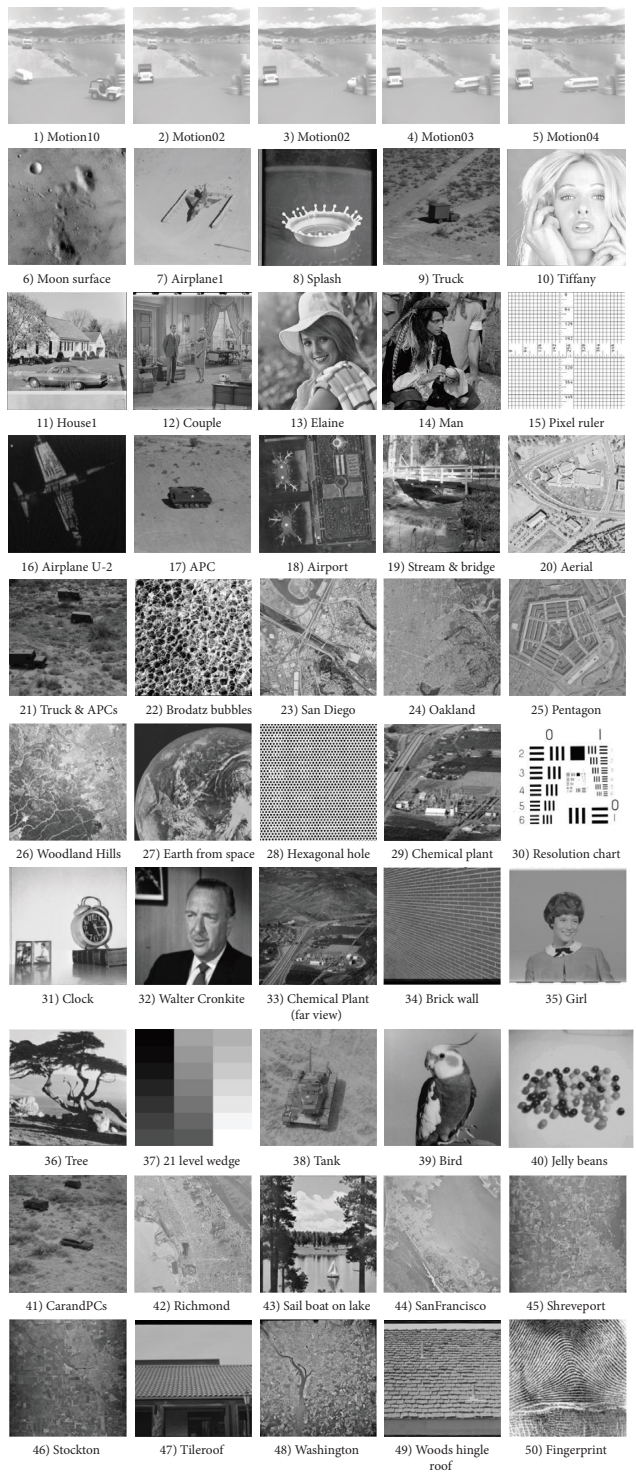


Figure 8: Fifty cover images.

dominance of the proposed method in terms of maximum embedding capacity. These results collectively demonstrate the effectiveness and superiority of the proposed RDH method in comparison to existing techniques.

Table 2 provides a comprehensive comparison of embedding capacity measured in bits per pixel among various IRDH methods, including those by Jung & Yoo [1], Zhang et al. [33], Lee & Huang [21], Tripathi and Prakash [27], and Bai et al. [28], alongside our proposed method. Our proposed approach achieves a remarkable embedding capacity of 4 bits per pixel, surpassing the average 2 bits per pixel offered by the existing methods. This substantial performance enhancement can be attributed to our unique technique of converting the decimal values of interpolated

Table 3: Evaluation of image quality and embedding capacity of the proposed method.

So. no.	Images	PSNR	Capacity
1	Motion10	29.06	1255383
2	Motion01	28.99	1240889
3	Motion02	28.60	1247241
4	Motion03	29.09	1304692
5	Motion04	29.16	1332763
6	Moon surface	28.93	1276002
7	Airplane1	28.76	1309917
8	Splash	28.56	1326826
9	Truck	28.62	1240835
10	Tiffany	29.12	1321975
11	House1	29.04	1291352
12	Couple	29.24	1226869
13	Elaine	29.16	1310048
14	Man	28.64	1262439
15	Pixel ruler	28.56	1329546
16	Airplane U-2	28.96	1330409
17	APC	28.90	1311793
18	Airport	29.08	1267324
19	Stream & bridge	29.28	1312364
20	Aerial	28.90	1253835
21	Truck & APCs	29.10	1270503
22	Brodatz grass	28.97	1239660
23	San Diego	28.99	1268294
24	Oakland	29.19	1292416
25	Pentagon	29.01	1275524
26	Woodland Hills	28.92	1284352
27	Earth from space	29.06	1308097
28	Hexagonal hole	28.61	1294324
29	Chemical plant	28.54	1283355
30	Resolution chart	29.26	1335265
31	Clock	29.20	1235580
32	Walter Cronkite	28.82	1232391
33	Chemical plant (far view)	29.15	1273532
34	Brick wall	28.63	1260188
35	Girl	28.93	1300964
36	Tree	28.61	1257173
37	21 level wedge	29.09	1272524
38	Tank	28.91	1299480
39	Bird	29.09	1311493
40	Jelly beans	29.26	1328842
41	CarandPCs	28.86	1253188
42	Richmond	29.06	1263608
43	Sail boat on lake	28.79	1334398
44	SanFrancisco	28.87	1297423
45	Shreveport	29.05	1285835
46	Stockton	28.96	1264508
47	Tileroof	28.46	1317791
48	Washington	29.04	1298894
49	Woods hingle roof	28.84	1315558
50	Fingerprint	28.80	1235855

pixels into logarithmic base 2 and subsequently replacing a specific number of bits with concealed information.

In order to thoroughly validate the effectiveness of the proposed method, a comprehensive study was conducted using hundreds of images. To avoid redundancy, we present the results of fifty representative images in Figure 8. Additionally, we recorded the corresponding outcomes of embedding capacity and PSNR with various threshold values for these images in Table 3. The analysis revealed that the highest embedding capacity achieved was 1,335,265 bits, while the lowest was 1,226,869 bits. Regarding PSNR, the highest value recorded was 29.28 dB, whereas the lowest was 28.46 dB. On average, the embedding capacity was found to be 1,284,870 bits and the PSNR averaged at 28.93 dB. These results lend further

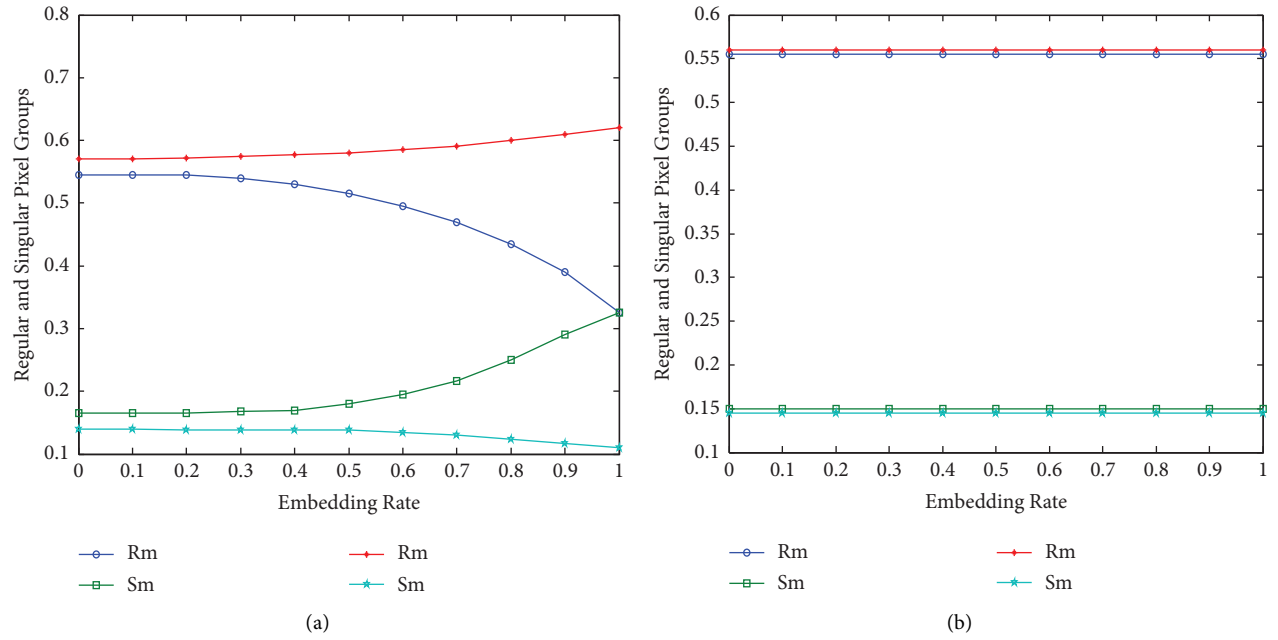


Figure 9: Stego image "Baboon" RS steganalysis. (a) Conventional LSB-embedding technique and (b) proposed scheme.

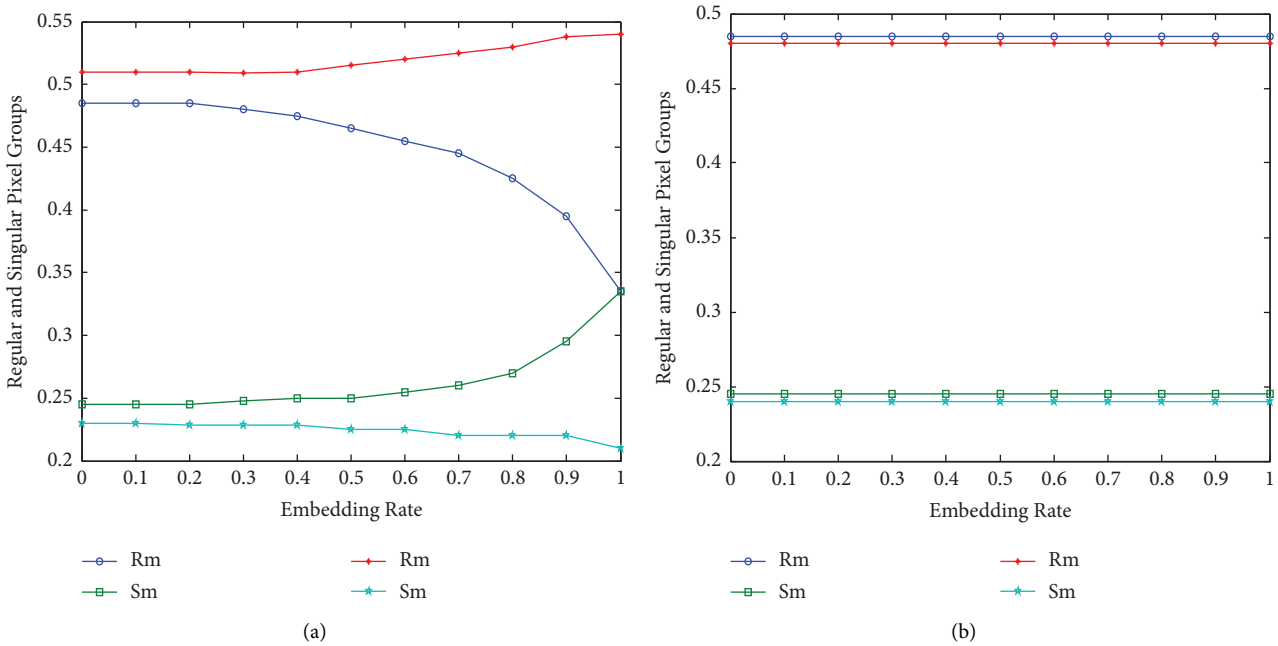


Figure 10: Stego image "Peppers" RS steganalysis. (a) conventional LSB-embedding technique and (b) proposed scheme.

support to the promising performance of the proposed method.

4.1. RS Steganalysis. In Figures 9 and 10, RS steganalysis is applied to the stego images "Baboon" and "Peppers" to evaluate both conventional LSB-embedding techniques and the proposed scheme. Fridrich's et al. [34] created RS steganalysis to find hidden data in steganographic images. We checked if the proposed scheme can be detected. RS steganalysis categorizes pixels into regular and singular groups. Successful detection relies on Rm ≅ R-m and Sm ≅ S-m. We used m and -m masks to assess hiding capacity. Conventional techniques are detectable, but the proposed scheme resists detection due to Rm ≅ R-m and Sm ≅ S-m. RS steganalysis cannot find secret data in the proposed scheme, making it resistant to detection. Results from Peppers and Baboon images support this conclusion in Figures 9 and 10. Experiments with all four input images yielded similar

results, so we present data from Peppers and Baboon images to support our conclusion and avoid redundancy.

5. Conclusion

A novel interpolation-based reversible data hiding approach is presented which aims to provide a high hiding capacity with decent image quality in terms of PSNR. The proposed method starts by using a modified neighbor mean interpolation to scale the original image, resulting in a higher-quality output. Then, a new technique for embedding secret data into the interpolated pixels is proposed. This is achieved by transforming the decimal value of the interpolated pixels into log base 2 and replacing a certain number of bits with the secret information bits. The result of this is a higher number of bits being embedded into the image while maintaining a high level of quality. Empirical results demonstrate that the proposed method can embed an average of 1,300,000 secret information bits with an average PSNR of 28.5 dB, outperforming previous and related works.

Data Availability

All the information will be available on request to Raju Pratap Sharma (rajupratapsharma@gmail.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. H. Jung and K. Y. Yoo, "Data hiding method using image interpolation," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 465–470, 2009.

[2] R. Kumar, K. K. Saini, and S. Chand, "A new steganography technique using snake scan ordering strategy," *The International Journal of the Image*, vol. 6, pp. 25–32, 2013.

[3] G. J. Simmons, "The prisoners' problem and the subliminal channel," *CRYPTO*, vol. 83, pp. 51–67, 1983.

[4] Y. T. Chang, C. T. Huang, C. F. Lee, and S. J. Wang, "Image interpolating based data hiding in conjunction with pixel-shifting of histogram," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1093–1110, 2013.

[5] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[6] J. Hu and T. Li, "Reversible steganography using extended image interpolation technique," *Computers & Electrical Engineering*, vol. 46, pp. 447–455, 2015.

[7] P. C. Mandal, I. Mukherjee, and B. N. Chatterji, "High capacity reversible and secured data hiding in images using interpolation and difference expansion technique," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3623–3644, 2021.

[8] F. S. Hassan and A. Gutub, "Novel embedding secrecy within images utilizing an improved interpolation-based reversible data hiding scheme," *Journal of King Saud University Computer and Information Sciences*, vol. 34, 2020.

[9] C. N. Yang, S. C. Hsu, and C. Kim, "Improving stego image quality in image interpolation based data hiding," *Computer Standards & Interfaces*, vol. 50, pp. 209–215, 2017.

[10] A. Malik, R. Kumar, and S. Singh, "A new image steganography technique based on pixel intensity and similarity in secret message," in *Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 828–831, Greater Noida, India, October 2018.

[11] A. Malik, S. Singh, and R. Kumar, "Recovery based high capacity reversible data hiding scheme using even odd embedding," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15803–15827, 2018.

[12] R. Kumar, A. Malik, S. Singh, B. Kumar, and S. Chand, "Reversible data hiding scheme for LZW codes using even-odd embedding strategy," in *Proceedings of the International Conference on Computing, Communication and Automation (ICCCA)*, pp. 1399–1403, Greater Noida, India, April 2016.

[13] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding: new paradigm in digital watermarking," *EURASIP Journal on Applied Signal Processing*, vol. 2, p. 185, 2002.

[14] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[15] R. Kumar, S. Chand, and S. Singh, "An improved histogram-shifting-imitated reversible data hiding based on HVS characteristics," *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13445–13457, 2018.

[16] R. Kumar, N. Kumar, and K. H. Jung, "I-PVO based high capacity reversible data hiding using bin reservation strategy," *Multimedia Tools and Applications*, vol. 79, no. 31-32, pp. 22635–22651, 2020.

[17] Y. Yalman, F. Akar, and I. Erturk, "An image interpolation based reversible data hiding method using R-weighted coding," in *Proceedings of the 2010 13th IEEE Int Conf Comput Sci Eng CSE*, pp. 346–350, Hong Kong, China, December 2010.

[18] R. Kumar, D. S. Kim, S. Lim, and K. H. Jung, "High-fidelity reversible data hiding using block extension strategy," *Proceedings of the 34th International Technical Conference on Circuits/Systems*, vol. 19, pp. 1–4, 2019.

[19] K. H. Jung, "A survey of interpolation-based reversible data hiding methods," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 7795–7810, 2018.

[20] S. R. Jan, S. J. Hsu, C. F. Chiu, and S. L. Chang, "An improved data hiding method using image interpolation," in *Proceedings of the 7th Int Conf Intell Inf hiding multimed signal process IIHMSP*, pp. 185–188, Dalian, China, October 2011.

[21] C. F. Lee and Y. L. Huang, "An efficient image interpolation increasing payload in reversible data hiding," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6712–6719, 2012.

[22] M. Fan, S. Zhong, and X. Xiong, "Reversible data hiding method for interpolated images based on modulo operation and prediction-error expansion," *IEEE Access*, vol. 11, pp. 27290–27302, 2023.

[23] A. Malik, G. Sikka, and H. K. Verma, "A reversible data hiding scheme for interpolated images based on pixel intensity range," *Multimedia Tools and Applications*, vol. 79, no. 25-26, pp. 18005–18031, 2020.

[24] X. Xiong, Y. Chen, M. Fan, and S. Zhong, "Adaptive reversible data hiding algorithm for interpolated images using sorting and coding," *Journal of Information Security and Applications*, vol. 66, Article ID 103137, 2022.

[25] Z. Huang, Y. Lin, and X. Chen, "A block-based adaptive high fidelity reversible data hiding scheme in interpolation domain," *Multimedia Tools and Applications*, vol. 2023, 2023.

[26] F. S. Hassan and A. Gutub, "Efficient reversible data hiding multimedia technique based on smart image interpolation," *Multimedia Tools and Applications*, vol. 79, no. 39-40, pp. 30087–30109, 2020.

[27] A. Tripathi and J. Prakash, "Blockchain enabled interpolation based reversible data hiding mechanism for protecting records," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 2023, 2023.

[28] X. Bai, Y. Chen, G. Duan, C. Feng, and W. Zhang, "A data hiding scheme based on the difference of image interpolation algorithms," *Journal of Information Security and Applications*, vol. 65, Article ID 103068, 2022.

[29] D. R. I. M. Setiadi, S. Rustad, P. N. Andono, and G. F. Shidik, "Graded fuzzy edge detection for imperceptibility optimization of image steganography," *The Imaging Science Journal*, vol. 5, pp. 1–13, 2023.

[30] 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 Processing*, vol. 206, Article ID 108908, 2023.

[31] S. R. Brahma, S. Singh, D. K. Gupta, and A. Malik, "A reversible data hiding technique using lower magnitude error channel pair selection," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 8467–8488, 2023.

[32] A. Malik, G. Sikka, and H. K. Verma, "An image interpolation based reversible data hiding scheme using pixel value adjusting feature," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13025–13046, 2017.

[33] X. Zhang, Z. Sun, Z. Tang, C. Yu, and X. Wang, "High capacity data hiding based on interpolated image," *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 9195–9218, 2017.

[34] J. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," in *Proceedings of the ACM workshop on multimedia and security*, pp. 27–30, Ontario, Canada, October 2001.