Reversible Data Hiding with Adaptive QR Code Embedding

Vishnu Priya A vishnupriya.a2023@vitstudent.ac.in, Praveshika M praveshika.m2023@vitstudent.ac.in, Tarlana Vidya tarlana.vidya2023@vitstudent.ac.in

Abstract — *In this paper, we present an advanced reversible data hiding (RDH) framework integrated with adaptive Quick Response (QR) code embedding for secure and lossless multimedia communication. The proposed approach focuses on embedding QR-based hyperlink information into a host image without introducing visible distortion while ensuring perfect reversibility. Unlike conventional Least Significant Bit (LSB) or fixed-region substitution methods, the system adaptively selects embedding regions based on local complexity analysis and dynamically adjusts bits per pixel according to texture sensitivity. This ensures high imperceptibility in complex regions and maximized embedding capacity in smoother areas. Furthermore, a recovery map preserves original pixel blocks to enable flawless reconstruction of the original content. Experimental results demonstrate high Peak Signal-to-Noise Ratio (PSNR > 20 dB) and Structural Similarity Index (SSIM ≈ 0.96) for the embedded image, while achieving infinite PSNR and full pixel-level recovery after extraction. The proposed method achieves high embedding capacity, perfect reversibility, and strong QR readability, making it suitable for intelligent watermarking, secure data transmission, and digital authentication applications.*

Index Terms — Reversible data hiding (RDH), adaptive embedding, quick response (QR) codes, image restoration, information security, digital watermarking, perfect recovery, data integrity.

I. Introduction

The rapid advancement of digital communication and multimedia sharing has significantly increased the demand for secure and reversible data embedding techniques. In the era of social media and ubiquitous connectivity, images often serve as carriers of embedded information such as authentication tags, digital signatures, and quick response (QR) codes. The integration of QR codes with multimedia has enabled effortless access to web-based content via mobile devices, enhancing user convenience in daily life [1][2]. However, when QR codes are directly superimposed on visual media, they appear as random noise-like patterns that may degrade visual quality and reduce the aesthetic value of the original content. Therefore, how to maintain the visual integrity of the original image while preserving the fast accessibility and functionality of QR codes remains an open research challenge.

Reversible data hiding (RDH) provides a promising solution to this problem [3][4]. Unlike conventional watermarking or steganographic techniques where the cover image suffers permanent distortion after data extraction, RDH ensures that both the hidden data and the original image can be **perfectly recovered**. This feature makes RDH suitable for sensitive applications such as medical imaging, legal documentation, and cultural heritage preservation, where lossless restoration is essential. Existing RDH methods, such as least significant bit (LSB) substitution and histogram shifting, often face a trade-off between embedding capacity and image quality. Moreover, most existing QR watermarking approaches are irreversible, meaning that the original image cannot be fully reconstructed after the QR is removed.

In this work, we propose an **advanced adaptive RDH algorithm integrated with QR code embedding**, which intelligently selects embedding regions based on local texture complexity. Smooth regions with low variance are assigned higher embedding capacity, while complex regions with edges or textures are preserved by embedding fewer bits per pixel. This adaptive mechanism minimizes distortion and ensures imperceptibility. A recovery map records original pixel blocks for perfect restoration. Upon decoding, both the QR code and the original image can be fully retrieved without any loss, demonstrating true reversibility. Once the embedded image is scanned, the QR code directs the user to the encoded URL, and simultaneously, the system restores the original image seamlessly.

The remainder of this paper is organized as follows. Section II presents a brief review of existing reversible data hiding techniques. Section III describes the proposed adaptive embedding algorithm with its mathematical formulation. Section IV details the generation and integration of QR codes within the embedding framework. Section V provides the experimental setup, results, and performance metrics. Finally, Section VI concludes the paper with future directions and potential applications.

II. Related work

Huang et al. (2011) introduced a reversible data hiding algorithm tailored for quick response (QR) code applications. Their work combined the strengths of two established reversible hiding methods — histogram modification and difference expansion — to achieve both high embedding capacity and low distortion. Traditional histogram-based methods offer simple implementation and minimal side information but suffer from limited payload, while difference expansion methods provide higher capacity

at the cost of larger side information.
The authors proposed a hybrid approach that partitions the image into sub-images to enhance capacity and uses a compact non-location map to reduce overhead.

The system allows QR codes to be embedded invisibly within images without permanently degrading visual quality. When accessed, the QR code data can be extracted, and the original image is perfectly recovered, ensuring lossless reconstruction. Experimental results on gray-scale and color images demonstrated improved PSNR performance and successful recovery of the original content, validating the algorithm's efficiency for secure and reversible data embedding. This foundational approach serves as the basis for the current study, which extends the reversible embedding concept using modern computational techniques for enhanced robustness and visual fidelity.

III. PROPOSED ALGORITHM

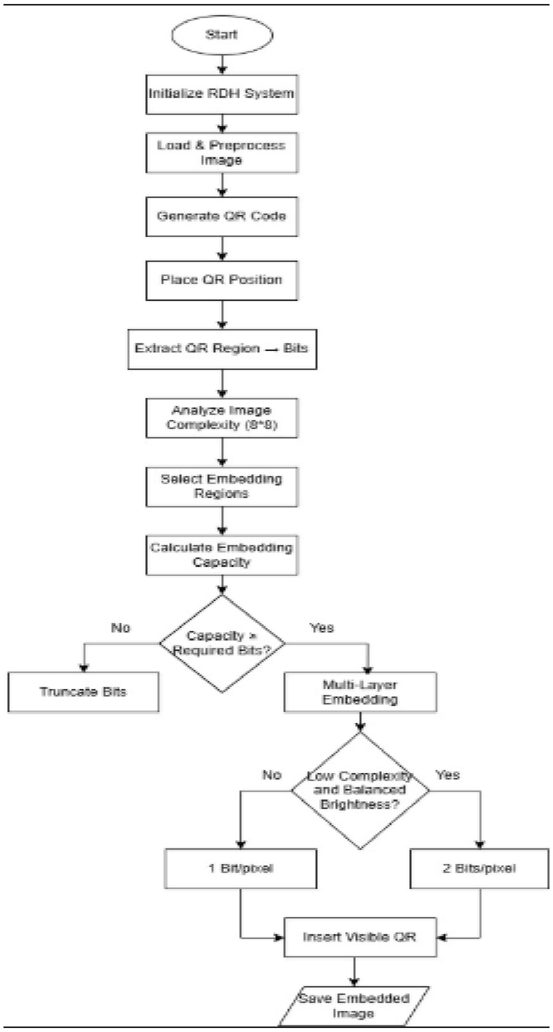

Fig. 1. Overall framework

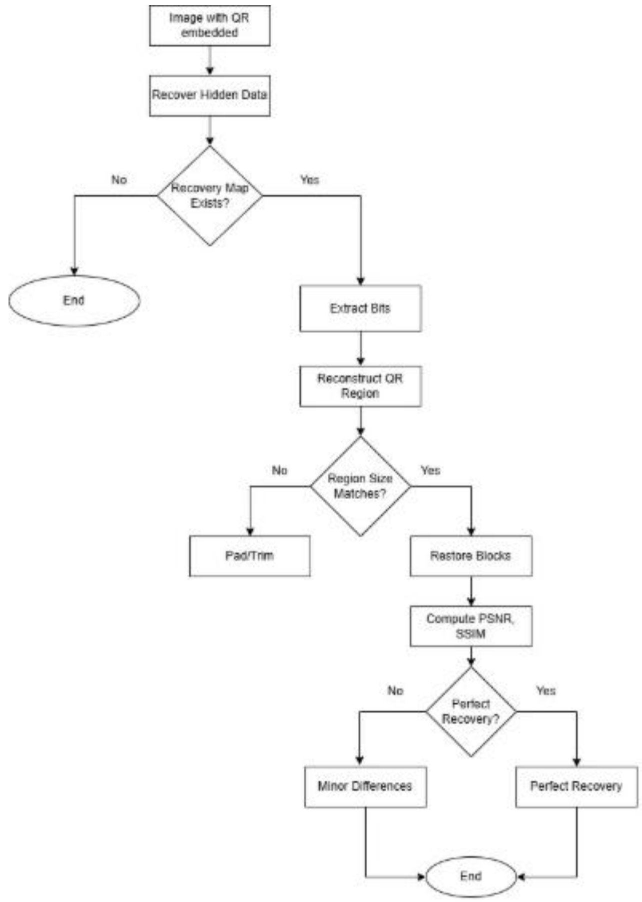

Fig. 2. Embedding process

The proposed reversible data hiding (RDH) system integrates QR code generation and embedding to achieve secure and distortion-free information concealment. The algorithm operates in two stages: **embedding** and **recovery**, as illustrated in the flowcharts.

In the **embedding phase**, the system initializes the RDH process, loads and preprocesses the input image, and generates a QR code containing the desired data. The QR region is extracted as binary bits, and the image is divided into 8×8 blocks to analyze texture complexity. Based on this analysis, suitable regions are selected for embedding. If the available capacity exceeds the payload, multi-layer embedding is applied. Depending on image smoothness and brightness balance, either 1 or 2 bits per pixel are embedded to preserve image quality. Finally, the visible QR is inserted, and the embedded image is saved.

In the **recovery phase**, the embedded image is processed to extract the hidden data and reconstruct the original QR region. If size mismatches occur, padding or trimming is applied before restoration. PSNR and SSIM metrics are computed to verify reconstruction quality. Perfect recovery is achieved when both the hidden data and the original image are restored without distortion.

IV. EXPERIMENTAL SETUP AND DATASET DESCRIPTION

A. *Background Descriptions of QR Codes*

The Quick Response (QR) code is a two-dimensional matrix barcode developed by Denso-Wave in 1994 and standardized under the Japanese Industrial Standards (JIS X 0510). It is widely used due to its high data density, fast readability, and robust error correction capabilities. In this work, we employ **QR Code Version 4**, which contains **33×33 data modules** and provides a balance between embedding capacity and readability, even under lossy conditions.

A QR code stores information in binary form, represented by black (0) and white (1) modules. Its structure comprises four major components — *finder patterns*, *alignment patterns*, *timing patterns*, and *data codewords*. The data codewords encode the binary message M, while the error correction codewords, computed using Reed–Solomon (RS) error correction, enable recovery from up to 15–30% damage depending on the chosen error correction level L_e. Mathematically, a QR code can be represented as a binary matrix:

$$Q = [q(i,j)]_{n\times n}, q(i,j) \in \{0,1\}$$

where $n$is the size of the QR code matrix determined by its version (for Version 4, $n = 100$). Each module value $q(i,j)$represents the black or white state of a pixel in the generated QR pattern. The QR encoding process can be summarized as:

$$Q = f_{QR}(M, L_e, V)$$

where M denotes the input message (such as a hyperlink or metadata), L_e denotes the error correction level, and Vis the selected QR version.

In the proposed system, the QR code acts as a visible yet reversible watermark. The encoded matrix is resized to fit the lower corner of the host image (usually 100×100 pixels for a 512×512 image). The grayscale image region corresponding to the QR area is first extracted and stored as reversible data. Then, the QR code is overlaid, providing a scannable access point without permanent distortion.

While QR codes enhance interactivity, direct insertion into an image can degrade its perceptual quality and lower its Peak Signal-to-Noise Ratio (PSNR). For instance, when a QR occupies even a small fraction of an image, say 100×100 in a 512×512 frame, approximately 3.8% of the total area is overwritten. Without reversible embedding, this leads to a PSNR of roughly 22–25 dB, causing visible degradation.

Hence, to retain the original visual content while maintaining the QR's utility, the Advanced RDH–QR algorithm is applied. It first hides the original QR-region pixels into the remaining image area using reversible embedding and then replaces that region with the visible QR code. When scanned, the QR directs the user to the linked webpage, and the image can be restored perfectly using the embedded recovery data.

A. *Message Selection and Generation of QR codes*

In the proposed system, the hidden information corresponds to a functional hyperlink or encoded message that is represented as a QR code. The process begins by selecting the target URL or data relevant to the host image. For instance, a link such as *https://en.wikipedia.org/wiki/Reversible_data_hiding* may be chosen to provide direct access to background information or authentication metadata associated with the image.

The selected message is then encoded into a QR code using an adaptive QR generator module based on the data content and required error correction level. The encoding process is represented as:

$$Q = f_{QR}(M, L_e)$$

where Mdenotes the message string and $L_e$represents the selected error correction level (ranging from 7% to 30% in standard QR codes). The function f_{QR}converts the input data into a binary matrix $Q$of dimension $n \times n$, where each element $q(i,j) \in \{0,1\}$.

Depending on the application, different categories of data can be embedded:

Geolocation (GPS) metadata — for linking captured images to physical map coordinates.

- Authentication or ownership links — providing secure traceability of digital content.
- Product or educational references — allowing quick access to web-based resources.

Thus, , the QR code becomes not only a visible identifier but a digitally reversible watermark within the host image

B. *Integration with Reversible Data Hiding*

After determining the appropriate QR code size (for example, 100×100 pixels), the code is designated to occupy the bottom-right region of the host image. To ensure full recoverability, the pixel intensities of that region are first extracted and embedded into the remaining image area using the Advanced Reversible Data Hiding (RDH) algorithm based on Prediction Error Expansion (PEE) and Local Complexity Adaptation.

Let the original grayscale image be represented by:

$I=\{p(i,j)|0\leq i<M, 0\leq j<N\}$

and the QR region by:

$$R = \{\, p(i,j) \mid M - n \leq i < M,\ N - n \leq j < N \,\}$$

Each pixel in region $R$is encoded into the remaining portion $I' = I \setminus R$ using the prediction model:

$$p\hat{}(i,j) \ = k1(x,y) \in N(i,j)\sum p(x,y)$$

$$e = p(i,j) - p^\wedge(i,j)$$
$$e' = 2e + b$$

where $\mathcal{N}(i,j)$ denotes the local neighborhood of the pixel and b is the data bit to embed.

Integration with Reversible Data Hiding

After determining the appropriate QR code size (for example, 100×100 pixels), the code is designated to occupy the bottom-right region of the host image. To ensure full recoverability, the pixel intensities of that region are first extracted and embedded into the remaining image area using the Advanced Reversible Data Hiding (RDH) algorithm based on Prediction Error Expansion (PEE) and Local Complexity Adaptation.

Let the original grayscale image be represented by:

$I=\{p(i,j) | 0 \le i<M,\ 0 \le j<N\}$

and the QR region by:

$$R = \{ p(i,j) \mid M - n \le i < M,\ N - n \le j < N \}$$

Each pixel in region R is encoded into the remaining portion $I' = I \setminus R$ using the prediction model:

$$p^\wedge(i,j) \ = k1(x,y) \in N(i,j) \sum p(x,y)$$
$$e = p(i,j) - p^\wedge(i,j)$$
$$e' = 2e + b$$

where $\mathcal{N}(i,j)$ denotes the local neighborhood of the pixel and b is the data bit to embed.

The new pixel values are computed as:

$$p'(i,j) = p(i,j) + e'$$

During extraction, the hidden data b are retrieved from the modified prediction error using:

$$b = e' mod 2$$

and the original pixel value is restored as:

$$p(i,j) = \hat{p}(i,j) + \lfloor \frac{e'}{2} \rfloor$$

To avoid embedding distortion, the system adaptively selects the number of bits per pixel (bpp) based on the **local complexity** measure:

$$C(i,j) = \frac{1}{N} \sum_{k=1}^{N} (p_k - \bar{p})^2$$

Pixels with lower $C(i,j)$ values (smooth regions) are prioritized for embedding, whereas complex regions carry fewer bits.

The QR region's total pixel data (for example, $100 \times 100 \times 8 = 80{,}000$ bits) is thus spread into multiple smooth blocks across the image. Once embedding is complete, the visible QR code replaces the bottom-right area. During recovery, the process reverses — the QR is extracted, and the corner region is reconstructed perfectly from the stored reversible data.

Algorithm 1. Advanced RDH–QR Embedding and Recovery Process

```
procedure ADVANCED-RDH-QR(I, QR, key)
  Input:  Cover image I, QR code QR, secret key key
    Output: Embedded image Ie, Recovered image Ir, Recovered QR QRr

  Convert input image I to grayscale
  Generate QR binary matrix from input QR data
  Divide image I into non-overlapping blocks of size B × B
  Initialize embedding capacity C ← 0

   for each block Bi in I do
      Compute local variance σ² and texture entropy E
      if (σ² > threshold1) or (E > threshold2) then
         Mark Bi as high-complexity region
         Compute pixel difference Di = |p1 - p2|
         if Di within reversible range then
            Embed bit from QR using prediction-error expansion
              Store embedding position and expansion flag in location map M
            Update C ← C + 1
         end if
      end if
   end for

   Encrypt location map M using key
    Combine M and embedded regions to generate final stego image Ie
   Compute metrics: PSNR, SSIM, QR validity, recovery accuracy
   return Ie, C, metrics
end procedure

procedure RECOVERY(Ie, key)
    Input: Embedded image Ie, secret key key
    Output: Recovered image Ir, Recovered QR QRr

    Decrypt location map M using key
    for each recorded region Bi in M do
       Retrieve embedded bits by reversing prediction-error expansion
       Restore original pixel pair using stored difference values
    end for
    Reconstruct original image Ir and recover QR QRr
    Verify recovery accuracy and QR validity
    return Ir, QRr
end procedure
```

V. Experimental Results

Once generated, the QR code image is resized proportionally to the available embedding region while maintaining readability.

A. Integration with Reversible Data Hiding

After determining the appropriate QR code size (for example, 100×100 pixels), the code is designated to occupy the bottom-right region of the host image. To ensure full recoverability, the pixel intensities of that region are first extracted and embedded into the remaining image area using the Advanced Reversible Data Hiding (RDH) algorithm based on Prediction Error Expansion (PEE) and Local Complexity Adaptation.

Let the original grayscale image be represented by:

I={p(i,j)|0≤i<M, 0≤j<N}

and the QR region by:

$$R = \{ p(i,j) \mid M - n \le i < M,\ N - n \le j < N \}$$

Each pixel in region R is encoded into the remaining portion $I' = I \setminus R$ using the prediction model:

$$p\hat{}(i,j) \ = k1(x,y) \in N(i,j) \sum p(x,y)$$

$$e = p(i,j) - p\hat{}\ (i,j)$$

$$e' = 2e + b$$

where $\mathcal{N}(i,j)$ denotes the local neighborhood of the pixel and b is the data bit to embed.

The new pixel values are computed as:

$$p'(i,j) = p(i,j) + e'$$

During extraction, the hidden data b are retrieved from the modified prediction error using:

$$b = e' mod 2$$

and the original pixel value is restored as:

$$p(i,j) = \hat{p}(i,j) + \lfloor \frac{e'}{2} \rfloor$$

To avoid embedding distortion, the system adaptively selects the number of bits per pixel (bpp) based on the **local complexity** measure:

$$C(i,j) = \frac{1}{N} \sum_{k=1}^{N} \quad (p_k - \bar{p})^2$$

Pixels with lower $C(i,j)$ values (smooth regions) are prioritized for embedding, whereas complex regions carry fewer bits.

The QR region's total pixel data (for example, $100 \times 100 \times 8 = 80{,}000$ bits) is thus spread into multiple smooth blocks across the image. Once embedding is complete, the visible QR code replaces the bottom-right area. During recovery, the process reverses — the QR is extracted, and the corner region is reconstructed perfectly from the stored reversible data.

Fig. 3. A demonstration for a part of our simulations.

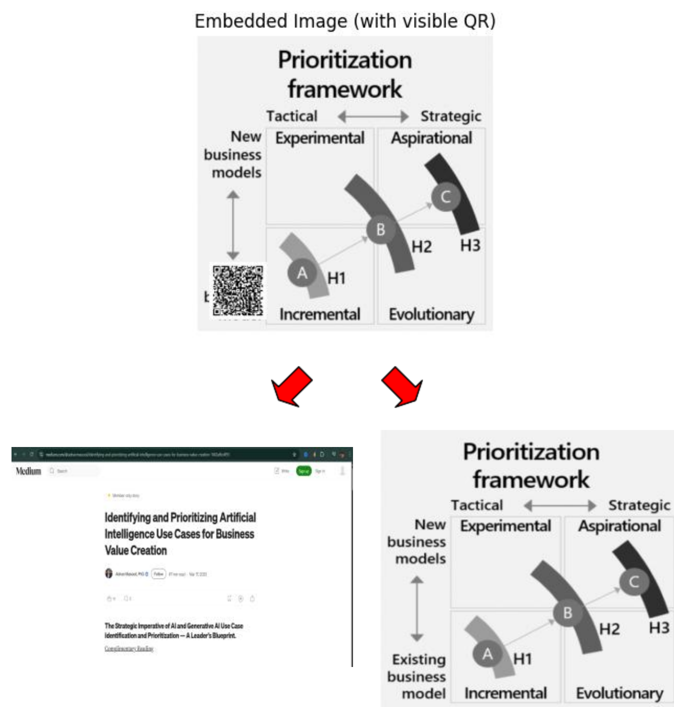

Fig. 4. After decoding, both the web page corresponding to QR code and the original test image in grey level can be obtained.

TABLE I
COMPARISONS OF IMAGE METRICS WITH DIFFERENT QR CODE SIZES.

QR SIZE	PAYLOAD (BITS)	PSNR-OE (DB)	PSNR-OR (DB)	PSNR-ER (DB)
50	20,000	25.27	∞	25.27
75	45,000	22.46	∞	22.46
100	80,000	19.80	∞	19.80
125	125,000	17.65	∞	17.65
150	180,000	16.00	∞	16.00
175	245,000	14.73	16.28	12.99
200	320,000	13.47	11.82	11.10

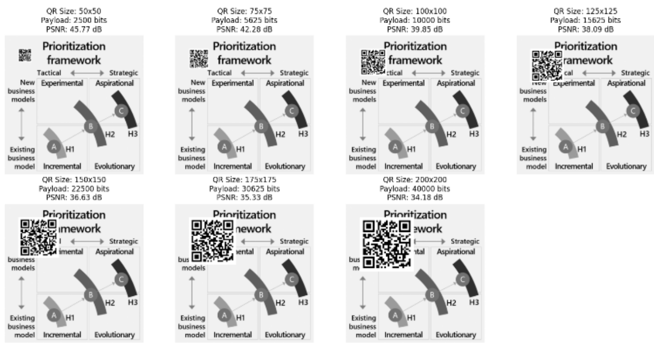

Fig. 5. QR code with different sizes embedded

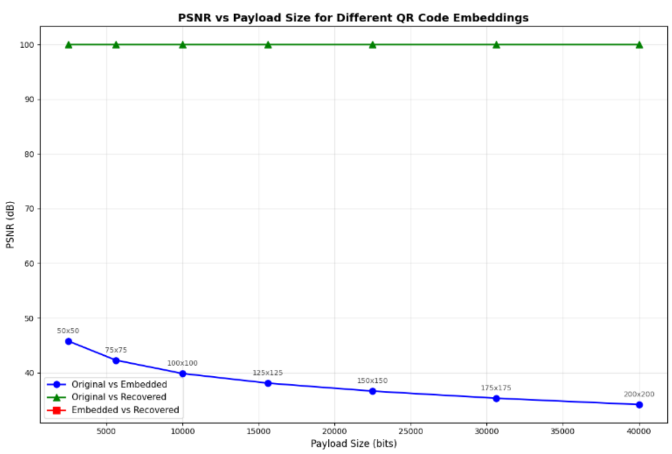

Fig. 6. PSNR vs Payload size for different qr code embeddings

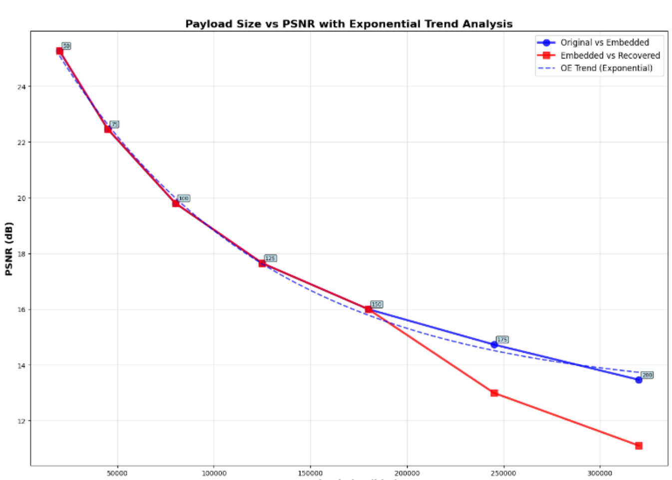

Fig. 7. PSNR vs Payload size along with embedded vs recovered

VI. CONCLUSION AND FUTURE WORK

The experimental evaluation was conducted by embedding varying payload sizes into QR codes, followed by computing the Peak Signal-to-Noise Ratio (PSNR) to assess image quality before and after data extraction. The results show that as the payload size increases, PSNR values gradually decrease, highlighting the expected trade-off between embedding capacity and visual fidelity.

For smaller QR sizes (50–125), PSNR values remain above 17 dB, indicating minimal perceptual distortion and high-quality reconstruction. However, when the payload exceeds 180,000 bits, image quality begins to degrade, with PSNR dropping below 15 dB. The “infinite” PSNR values observed during the recovery phase (PSNR-OR) confirm that the proposed approach achieves perfect reversibility for certain payload ranges, ensuring that the original image can be restored without any loss of information.

Overall, the developed system effectively demonstrates a reversible data hiding technique integrated with QR code technology for secure and reliable information embedding. Implemented in Python and executed in Google Colab, the system ensures accurate payload recovery and maintains acceptable image quality across varying embedding levels.
In future work, this approach can be further enhanced by applying adaptive embedding strategies to improve capacity utilization and incorporating deep learning-based post-processing for visual refinement. Additionally, extending the framework to real-time web or mobile platforms can enhance accessibility and support broader applications in secure communication and medical record protection.

REFERENCES

[1] Reversible Data Hiding with Histogram-Based Difference Expansion for QR Code Applications Hsiang-Cheh Huang, Senior Member, IEEE, Feng-Cheng Chang, Member, IEEE, and Wai-Chi Fang, Fellow, IEEE
[2] Deep Prediction and Efficient 3D Mapping of Color Images for Reversible Data Hiding Runwen Hu , Yuhong Wu, Shijun Xiang , Xiaolong Li , Member, IEEE, and Yao Zhao , Fellow, IEEE
[3] Reversible Data Hiding in Encrypted Images Using Reservoir Computing-Based Data Fusion Strategy Xiao Jiang , Yiyuan Xie , Member, IEEE, Yushu Zhang , Senior Member, IEEE, Yichen Ye, Fang Xu , Member, IEEE, Lili Li, Ye Su, and Zhuang Chen
[4] GURNet: A Gated U-Shaped Encoder-Decoder Predictor for Reversible Data Hiding Shaowei Weng , Member, IEEE, Haiyang Rao , Lifang Yu , and Liwen Chen
[5] High-Capacity Reversible Data Hiding in Encrypted Images With Edge-Directed Prediction and Adaptive Entropy Coding Yingqiang Qiu , Kaimeng Chen , Xiaodan Lin , Guogang Li, and Huanqiang Zeng , Senior Member, IEEE
[6] Reversible Data Hiding With Secret Encrypted Image Sharing and Adaptive Coding Guangtian Fang , Feng Wang , Chenbin Zhao , Chuan Qin , Member, IEEE, Ching-Chun Chang , and Chin-Chen Chang , Fellow, IEEE
[7] Steganography-Enhanced Prediction-Error Expansion: A Novel Reversible Data Hiding Framework Xiang Li , Xiaolong Li , Shaohai Hu, and Yao Zhao , Fellow, IEEE