

LINEAR PREDICTIVE MODELING AND ADVANCED PREDICTION-ERROR CORRECTION FOR REVERSIBLE WATERMARKING

THANET KETTHONG¹ AND LUCKSANA ROMYASAMIT^{2,*}

¹Department of Electrical Engineering

²Department of Computer Science

Faculty of Science and Technology

Dhonburi Rajabhat University

172 Itsaraphap Road, Wat Kanlaya Subdistrict, Thonburi District, Bangkok 10600, Thailand

Thanet.K@dru.ac.th; *Corresponding author: Lucksana.r@dru.ac.th

Received October 2025; revised February 2026

ABSTRACT. *Digital watermarking, particularly reversible watermarking, has become crucial in protecting intellectual property and ensuring data integrity in various domains, such as medical and military applications. The design of efficient watermarking algorithms remains a challenge, particularly in balancing embedding capacity with image quality. Current reversible watermarking methods often face limitations in embedding capacity and image distortion due to large Prediction-Error Adjustment (PEA) maps and insufficient optimization of embedding techniques. This paper introduces a novel approach based on linear predictive modeling and the Advanced Prediction-Error Correction (APEC) map. The APEC map enhances the PEA map reduction process by using two distinct sets of pixels, enabling the image receiver to determine embedding formulas based solely on pixel value properties, eliminating the need for a PEA location map. The proposed method significantly improves the embedding capacity and image quality, particularly in images with high pixel value variation. Comparative analysis shows that the linear predictive function outperforms other methods, yielding the highest PSNR values for the same payload size, particularly in the second watermark layer. The method also reduces the size of the PEA map, increasing the maximum information-hiding capacity while maintaining the target PSNR. This work presents a significant advancement in reversible watermarking by optimizing the predictive model and enhancing prediction-error correction strategies. The proposed approach demonstrates superior performance in high-noise conditions, making it suitable for specialized applications such as medical imaging and satellite imagery. Future work will focus on further improving prediction accuracy, especially for images with low pixel value variation.*

Keywords: Reversible watermarking, Expansion embedding, Prediction error, Linear predictive modeling, Prediction error correction

1. **Introduction.** Information hiding, steganography, and digital watermarking are three interrelated research fields that share a wide range of technical foundations and methodologies. Despite their overlap, they differ in objectives and design philosophies. Steganography primarily aims to conceal the existence of information, whereas digital watermarking focuses on embedding auxiliary data into digital media, such as images, audio, or video, for purposes including copyright protection, content authentication, and broadcast monitoring. These differences have shaped the evolution of algorithms and application domains, leading to significant advancements in both reversible and irreversible techniques [1]. Recent research has expanded the scope of watermarking methods. For instance, Fan

et al. [2] introduced a robust non-blind watermarking scheme leveraging low-frequency construction techniques. Their method modifies pixel values based on histogram analysis and optimal pixel weights, effectively minimizing noise and enhancing resistance to various attacks. Similarly, Meng et al. [3] presented an adaptive watermarking algorithm for copyright protection that employs an improved Arnold permutation for encryption and adapts embedding positions within high-frequency sub-bands, thereby bolstering robustness and supporting multi-size watermark embedding.

In the context of deep learning, Zhong et al. [4] conducted a comprehensive survey of deep learning-based image watermarking techniques, categorizing them into embedder-extractor, feature transformation, and hybrid methods, which provides valuable insights for future research in this area. More recently, advancements in watermarking have explored its application in emerging domains such as 3D scene protection. Liang and Li [5] proposed a multi-view graph deep learning system for 3D model retrieval based on a fusion Vision-Transformer architecture, demonstrating the effectiveness of combining graph neural networks with transformer-based feature extraction. Similarly, Li [6] introduced a dynamic network representation learning method that integrates graph neural networks with temporal information, highlighting the importance of adaptive feature learning in complex data environments. These studies underscore the growing role of advanced neural architectures in multimedia information processing, which aligns with the increasing interest in learning-based approaches for watermarking applications. Jachak et al. [7] further reinforced the importance of watermarking, noting its role in protecting the authenticity and integrity of digital files. They emphasized the development of both robust and visible techniques that withstand diverse attacks, yet acknowledged that not all watermarking methods can offer reversible recovery of original images, particularly in cases that demand full data retrieval.

The evolution of reversible watermarking has drawn significant attention, particularly for applications where the original data must be recoverable. Such use cases include medical and military applications, where data integrity and recovery are paramount. Early reversible watermarking techniques [8-11] often relied on modulo-arithmetic-based additive spread spectrum methods, which introduced salt-and-pepper artifacts, complicating the watermark retrieval process. De Vleeschouwer et al. [12] addressed these artifacts by leveraging bijective transformations of the image histogram, while other researchers like Fridrich et al. [13] employed lossless bit-plan compression techniques to enhance the embedding area. The field progressed with novel methods such as Xuan et al.'s [14] histogram modification operation, which effectively prevented overflow or underflow issues caused by high-frequency coefficient modification before embedding the watermark. In more recent advancements, researchers have proposed various innovative techniques to further improve capacity and distortion-free recovery. These include methods such as the expansion embedding technique [15], prediction-error adjustment [16], and histogram shifting [17] to enhance data compression and recovery accuracy. Thodi and Rodriguez [18] built upon these ideas by introducing more advanced histogram shifting and overflow mapping strategies to improve the efficiency of prediction-error-based watermarking, further enhancing data compression while maintaining higher recovery fidelity. Weng et al. [19] expanded upon Thodi and Rodriguez [18] methods, optimizing the use of prediction-error adjustments and further improving data compression, recovery accuracy, and overall image quality. Huang [20] introduced a reversible visible watermarking approach for BTC-compressed images, embedding visible watermark bits in designated areas while ensuring distortion-free recovery of the original image. Additionally, Tang et al. [21] and Huang et al. [22] introduced improved prediction error techniques and histogram-shifting-based methods to

increase capacity and accuracy for data hiding, particularly in encrypted and compressed image domains.

The application of Reversible Data Hiding (RDH) continues to evolve, with recent work focusing on encrypted image protection and blockchain integration. Rai et al. [23] proposed a high-capacity RDH scheme based on a two-layer embedding strategy combined with blockchain technology for traceability and data integrity. Similarly, Panchikkil et al. [24] utilized Fibonacci transforms and ensemble learning techniques to achieve effective image recovery in encrypted settings, further advancing the field. Tahiri et al. [25] proposed an enhanced reversible watermarking system for color images using Quaternary Charlier Moments (QCM) and Discrete Wavelet Transform (DWT), which combined with QR/SVD, Henon chaotic maps, and an arithmetic optimization algorithm, and enhanced the robustness, capacity, and security of reversible watermarking systems. Qiu et al. [26] introduced a scheme that employs an improved Convolutional Neural Network (CNN) predictor, leveraging feature, pixel, and complexity prediction modules to achieve improved rate-distortion performance for reversible watermarking. Ragab et al. [27] provided a thorough review of recent advancements in image steganography and RDH techniques, highlighting the challenges posed by attacks such as steganalysis and evaluating the effectiveness of techniques in both encrypted and compressed domains. Chen et al. [28] further developed the concept of Deep Robust Reversible Watermarking (DRRW), proposing a deep learning-based approach to enhance the robustness and reduce complexity in reversible watermarking, achieving lossless recovery while significantly reducing auxiliary bitstream by $43.86\times$.

In summary, while the landscape of reversible watermarking and data hiding has seen substantial advancements, challenges remain in balancing high capacity, robust protection, and lossless recovery. The contributions of this paper lie in extending current techniques by employing linear predictive modeling and advanced prediction-error correction strategies. These methods aim to enhance the robustness, capacity, and security of reversible watermarking systems, providing a novel approach for tackling some of the key challenges identified in the literature.

This paper consists of six sections. After an introduction given in Section 1, the related works of linear predictive modeling are presented in Section 2. Details of data embedding are illustrated in Section 3. The data extraction and image recovery is described in Section 4. Simulation and experimental results are illustrated in Section 5, while conclusions and future work are given in Section 6.

2. Linear Predictive Modeling. The optimization of this linear predictor's design for watermark embedding, achieved via linear predictive modeling, can be shown in the block diagram of Figure 1. Let x_1^i, x_2^i, x_3^i and x_4^i be independent variables representing neighborhood pixel values of any position in an image. y^i is a dependent variable representing the position's pixel value. Let I be a set of pixel values in an image.

$$x_1^i, x_2^i, x_3^i, x_4^i, y^i \in I, \quad i = 1, 2, \dots, (r-1)(c-2)$$

where r is a number of row and c is a number of columns in an image I . We want to find a function which approximates a set of values in I by using function $f(x_1, x_2, x_3, x_4)$ which represents the relation between the neighborhood pixels (x_1, x_2, x_3, x_4) and the target pixel y .

The linear predictive function can be written in the form:

$$f(x_1, x_2, x_3, x_4) = c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \quad (1)$$

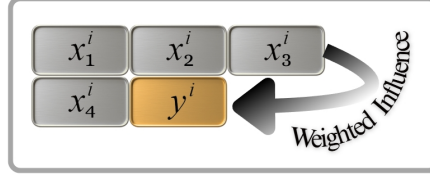


FIGURE 1. Linear predictive modeling

Assuming that the function specified in Equation (1) consists of four terms and four unknown coefficients, we can determine these coefficients by solving a system of linear equations. For each pixel position i , we have one equation:

$$y^i = (c_1 x_1^i + c_2 x_2^i + c_3 x_3^i + c_4 x_4^i), \quad i = 1, 2, \dots, N$$

where $N = (r - 1)(c - 2)$ is the total number of valid pixel neighborhoods in the image. This yields an overdetermined system of N equations with only four unknowns.

2.1. Coefficient estimation using least squares. In the proposed formulation, the number of available observations N is typically much larger than the number of unknown coefficients (i.e., $N \gg 4$). Consequently, the resulting system of equations is overdetermined, and it is generally impossible to obtain a solution that satisfies all equations simultaneously. To determine an appropriate set of coefficients, the problem is formulated as a least-squares optimization task. Specifically, we estimate the coefficient vector

$$c = [c_1, c_2, c_3, c_4]^T$$

such that the total squared prediction error across all neighborhoods is minimized. The objective function can be written as

$$E = \sum_{i=1}^N (y_i - c^T x_i)^2$$

where $x_i = [x_{1i}, x_{2i}, x_{3i}, x_{4i}]^T$ represents the vector of neighboring pixel values associated with the i -th sample, and y_i denotes the corresponding target pixel value [29].

This formulation corresponds to a classical linear least-squares problem. By differentiating the objective function E with respect to the coefficient vector c and setting the derivative equal to zero, the following normal equation is obtained:

$$F^T F c = F^T Y$$

In this representation, F denotes an $N \times 4$ design matrix whose i -th row is given by x_i^T , while Y is an $N \times 1$ vector containing the target pixel values, expressed as $Y = [y_1, y_2, \dots, y_N]^T$. Solving the normal equation yields the optimal coefficient vector in closed form as

$$c = (F^T F)^{-1} F^T Y \quad (2)$$

2.2. Efficient implementation for full-size images. For a typical image resolution of 512×512 pixels, the number of valid pixel neighborhoods N can reach approximately 260,000. Explicitly constructing the feature matrix F would require storing more than one million floating-point values, resulting in high memory consumption and reduced efficiency, particularly for large images or repeated watermark embedding operations. To address this issue, the proposed method avoids explicitly forming F . Instead, the matrices $F^T F$ and $F^T Y$ are computed through an incremental accumulation process. During a single pass through the image, the contributions from each neighborhood are progressively accumulated as follows:

$$F^T Y = \sum_{i=1}^N x_i y_i$$

In this formulation, $x_i x_i^T$ denotes the outer product of the feature vector for the i -th neighborhood, producing a 4×4 matrix, while $x_i y_i$ represents a 4×1 vector obtained by scaling the feature vector with the corresponding target value y_i . These quantities are accumulated iteratively through simple addition during the image scanning process. This strategy improves both memory and computational efficiency. Only two small matrices, a 4×4 matrix and a 4×1 vector, need to be stored regardless of image size. Moreover, the accumulation process has linear complexity $O(N)$, making it computationally lightweight relative to the overall watermark embedding procedure. After accumulation, the coefficient vector c is obtained by solving the resulting 4×4 linear system in Equation (2).

2.3. Computational complexity and numerical stability.

2.3.1. Computational complexity. The computational burden associated with estimating the coefficient vector is primarily determined by the incremental accumulation stage. Since each valid pixel neighborhood contributes once to the summation, the total computational complexity grows linearly with the number of neighborhoods N , resulting in a complexity of $O(N)$. After the accumulation process is completed, the coefficient vector is obtained by solving a 4×4 linear system. The cost of solving such a small system using standard numerical techniques, such as LU decomposition, is constant and does not depend on the value of N . Consequently, the computational effort required for coefficient estimation is minimal when compared with the pixel-wise operations involved in the watermark embedding and extraction stages.

2.3.2. Numerical stability. It may occur if the matrix $F^T F$ becomes ill-conditioned, which can arise when predictor variables representing neighboring pixel intensities are highly correlated. Such multicollinearity can affect the stability of solutions obtained through direct inversion of the normal equation. To improve robustness, alternative least-squares methods can be employed. QR decomposition solves the least-squares problem without explicitly forming $F^T F$, thereby reducing numerical instability. Singular Value Decomposition (SVD) provides an even more stable solution when the matrix is near singular, although with slightly higher computational cost. In this work, $F^T F$ is only a 4×4 matrix, so the computational overhead of these methods is negligible. For efficiency, LU decomposition is used in the current implementation, while QR or SVD can be adopted when higher numerical stability is required.

3. Data Embedding.

3.1. Prediction-error expansion. The embedding process involves computing PE from neighborhood pixels and embedding the information bit in the expanded PE which is defined as a difference between pixel value (a) and its prediction value ($\hat{a}$) that is set to be the even integer less than or equal to $\hat{a}$, $\hat{a} = 2 \lfloor \frac{\hat{a}}{2} \rfloor$.

$$p = a - \hat{a} \quad (3)$$

Then we expand PE to make a vacant bit at the end of a bit string and add an information bit w where $w \in \{0, 1\}$.

$$p_w = 2p + w \quad (4)$$

The result of watermarked pixel value is

$$a_w = \hat{a} + p_w \quad (5)$$

The information bit is extracted from LSB (Least Significant Bit) of p_w and can restore an original p as

$$p = \left\lfloor \frac{p_w}{2} \right\rfloor \quad (6)$$

3.2. Prediction-error shifting. The threshold T is determined by the desired embedding payload and acceptable distortion. Empirically, it is selected by analyzing the prediction-error distribution and choosing the smallest value that provides sufficient expandable pixels. Analytically, T reflects a trade-off between embedding capacity and image quality. Larger values increase payload capacity but reduce PSNR because pixels with higher prediction errors introduce greater distortion. This trade-off is confirmed in Table 1, where increasing T from 5 to 20 increases payload capacity while reducing PSNR across different image types, including medical and natural images. Based on this principle, the proposed method uses T to control the expansion process. To limit distortion, only prediction-errors satisfying $|p| \leq T$ are selected for expansion and embedding. The expanded prediction-error satisfies

$$\begin{aligned} -T &\leq p \leq T \\ -2T &\leq 2p + w \leq 2T + 1 \end{aligned}$$

To differentiate the expanded PE from unchanged one, we shift prediction-error p that $|p| > T$ as follows [18]:

$$p_s = p + \text{sign}(p) \cdot (T + 1) \quad (7)$$

where p_s is shifted prediction-error and $\text{sign}(p)$ is 1 if $p > 0$, 0 if $p = 0$ and -1 if $p < 0$.

For p that is not expanded, we just embedded information bit by replacing it on the prediction-error's LSB.

$$p_w = 2 \left\lfloor \frac{p_{NE}}{2} \right\rfloor + w \quad (8)$$

where unexpanded prediction-error, p_{NE} , can be both shifted and unshifted prediction-error.

Since LSBs of p_{NE} will be replaced by the watermark, we need to collect these LSBs for image recovering and embedded p_{NE} can be conversed back to original value using

$$p_{NE} = \begin{cases} \hat{p} + LSB - \text{sign}(\hat{p})(T + 1), & p \text{ is shifted} \\ \hat{p} + LSB, & p \text{ is unshifted} \end{cases} \quad (9)$$

where $\hat{p}$ is a p_w after extracting its LSB, $\hat{p} = 2 \left\lfloor \frac{p_w}{2} \right\rfloor$.

3.3. Advanced prediction-error correction. Let p be any prediction error in images. The PEA map can be represented by Figure 2, and PEA function is defined as [19]

$$M(p) = \begin{cases} 2p + w & \text{for } |p| \leq T \\ p + \text{sign}(p) \cdot (T + 1) & \text{for } |p| > T \end{cases} \quad (10)$$

PEA location map is defined as

$$L(p) = \begin{cases} 1 & \text{if } M(p) \in [-\hat{a}, 255 - \hat{a}] \\ 0 & \text{if } M(p) \notin [-\hat{a}, 255 - \hat{a}] \end{cases} \quad (11)$$

Definition 3.1. We say that a prediction error p is expandable if $2p + w \in [-\hat{a}, 255 - \hat{a}]$ for both $w = 0, 1$.

Definition 3.2. We say that a prediction error p is shiftable if $p + \text{sign}(p) \cdot (T + 1) \in [-\hat{a}, 255 - \hat{a}]$.

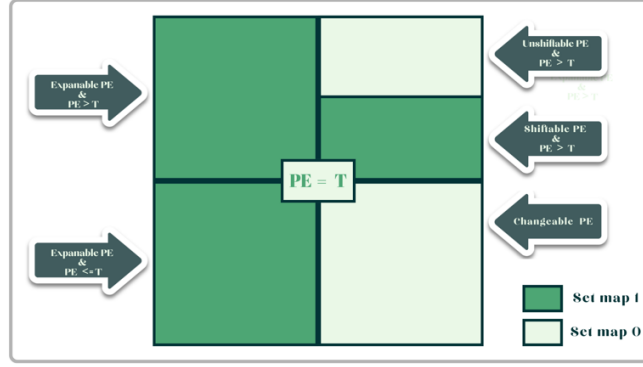


FIGURE 2. PEA map

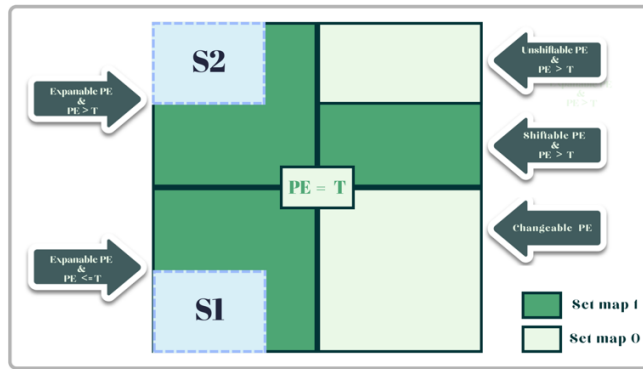


FIGURE 3. APEC map

We propose an enhanced approach called Advanced Prediction-Error Correction (APEC) map to reduce the size of the Prediction-Error Adjustment (PEA) map, as shown in Figure 3. This reduction is achieved by utilizing two distinct sets of pixels, where the image receiver can determine the corresponding embedding formula based solely on the pixel value properties, rather than relying on the PEA location map. These two sets can be considered as

Let S_1 be a set of pixel which $|\hat{p}| \leq 2T$ and $\hat{p} + w$ is expandable for both $w = 0, 1$, when $\hat{p} = 2 \left\lfloor \frac{p_w}{2} \right\rfloor$. (12)

Let S_2 be a set of pixel which $|\hat{p}| > 2T$ and $\hat{p} + w$ is shiftable for both $w = 0, 1$, when $\hat{p} = 2 \left\lfloor \frac{p_w}{2} \right\rfloor$. (13)

All watermarked pixels in these two sets are set to be 1 in the PEA location map and can be easily checked both by the sender and receiver. We get rid off these pixels from PEA location map to reduce information embedded into an image which can decrease the embedded load. Next, a reduced map is compressed using a lossless compression. A bit stream embedded into image includes a compressed reduced PEA map, LSB and watermark. The procedures of the proposed APEC map and watermark embedding can be summarized and represented by the pseudo code as shown in Figure 4, while the algorithm of the proposed APEC map and watermark embedding can be visualized by the flowchart diagram shown in Figure 5.

Figure 4 illustrates the comprehensive pseudo code for the proposed reduced APEC map and watermark embedding procedures. This algorithm outlines a ten-step process

reduced APEC map and watermark embedding procedures:

Step 1: Initialize the image (a), predicted pixel values ($\hat{a}$), and parameters for the algorithm.

Step 2: Calculate the prediction error: $p = a - \hat{a}$.

Step 3: If $p \leq T$, then:
 If p is expandable, expand p
 Else, keep p unchanged
 Else:
 If p is shiftable, shift p
 Else, keep p unchanged.

Step 4: Create Map: If p is expandable or shiftable, set map = 1, otherwise map = 0.

Step 5: Reduce Map: If p falls into set S_1 or S_2 , remove it from the map.

Step 6: Collect LSB (Least Significant Bit) for watermark embedding.

Step 7: Construct the Bitstream: Header + Map + LSB + Watermark.

Step 8: Embed the Bitstream: Insert the bitstream at the least significant bit of p to form p_w .

Step 9: Generate the watermarked image: $a_w = \hat{a} + p_w$.

Step 10: Output the watermarked image a_w .

FIGURE 4. Pseudo code of the proposed reduced APEC map and watermark embedding procedures

designed to achieve high embedding capacity while maintaining image quality. The procedure commences with Step 1, where the algorithm initializes the original image (a), the predicted pixel values ($\hat{a}$), and all necessary working parameters. Following initialization, Step 2 calculates the prediction error (p) for each pixel by finding the difference between the original and predicted values, as defined by the equation $p = a - \hat{a}$. The core logic for error adjustment is detailed in Step 3, which involves two main decision branches based on a predefined threshold (T). The process first checks if the absolute value of the prediction error (p) falls below or equals the threshold $p \leq T$.

- If this condition is met, the algorithm determines if the error (p) is expandable. If p is expandable, the value is expanded; otherwise, it remains unchanged.
- If the initial threshold condition is not met (Else), the process checks for shiftability. If the error (p) is shiftable, the value is shifted; otherwise, it remains unchanged.

The procedure then moves to Step 4, where the reduced APEC map (referred to simply as ‘Map’ in the figure) is created. The map bit is set to 1 if the prediction error (p) is determined to be either *expandable* or *shiftable*, signifying an area where embedding is possible. Conversely, the map bit is set to 0 (otherwise), indicating an unmodifiable area. The algorithm introduces a crucial size reduction technique in Step 5: Reduce Map. If the prediction error (p) belongs to a specific set of determined states, S_1 or S_2 , the corresponding pixel is removed from the map. This systematic removal optimizes the map, directly contributing to the final high embedding capacity. Subsequently, Step 6 collects the Least Significant Bit (LSB) from the appropriate pixels. This collected LSB data is necessary to ensure the reversibility of the method and is incorporated into the final bitstream. In Step 7, the comprehensive bitstream is constructed. This bitstream comprises the Header (which contains necessary auxiliary data for extraction), the Reduced APEC Map, the Collected LSBs, and finally, the Watermark data itself. Step 8 then performs the final embedding: the constructed bitstream is inserted into the LSB of the adjusted prediction error (p) to form the watermarked prediction error, denoted p_w .

Finally, the watermarked image is generated in Step 9. This involves adding the watermarked prediction error (p_w) back to the predicted pixel value ($\hat{a}$), as expressed by the formula $a_w = \hat{a} + p_w$. Step 10 simply outputs the resulting watermarked image a_w . This complete set of steps defines the entire embedding process.

A schematic overview of the proposed method is provided in Figure 5, outlining the process for constructing the reduced APEC map and executing the watermark embedding procedure. The entire process is divided into three main phases: coefficient calculation,

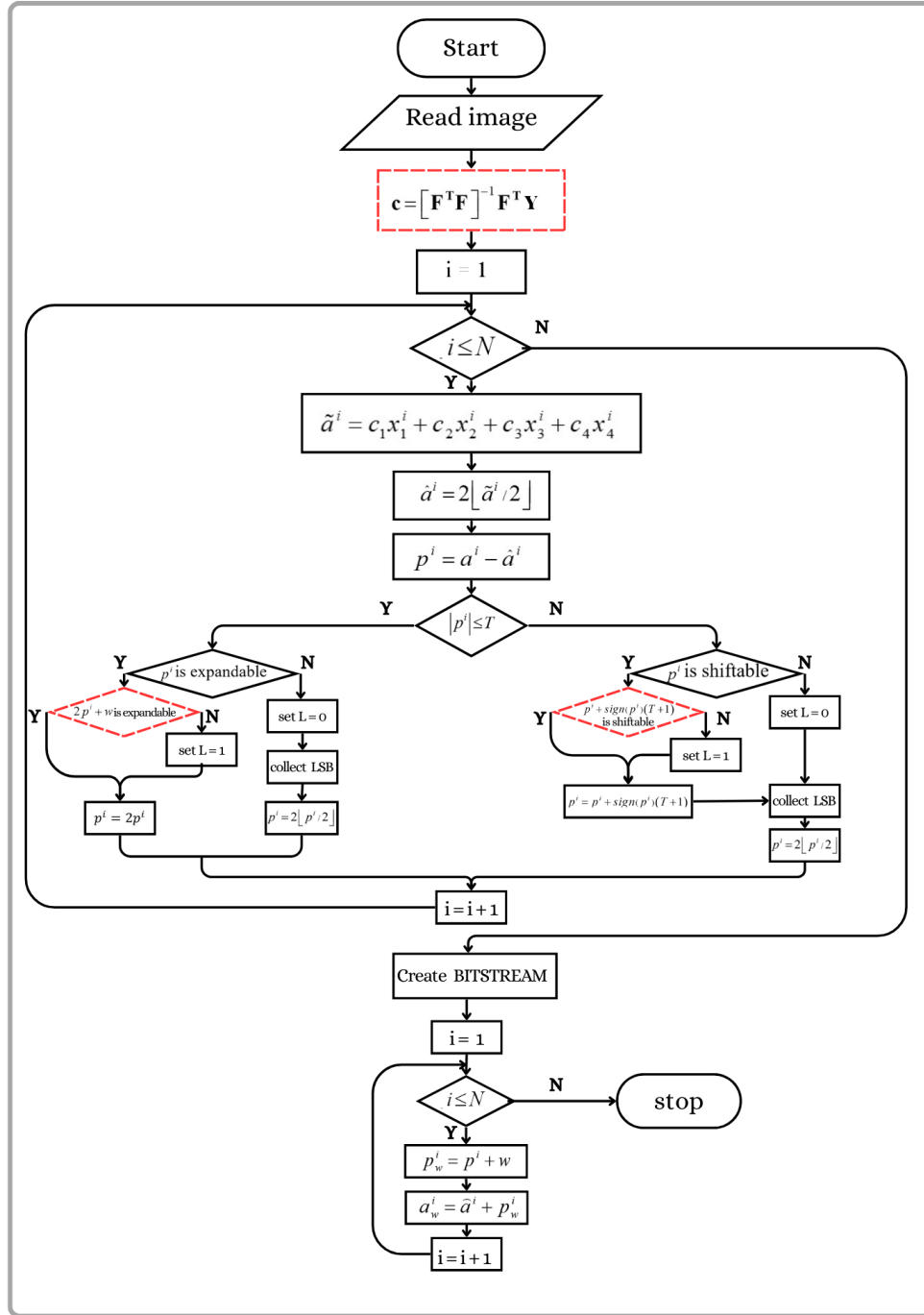


FIGURE 5. Schematic overview of the proposed integrated framework depicting the complete workflow comprising reduced APEC map construction and watermark embedding

pixel-wise error adjustment, and final bitstream embedding. The procedure initiates by reading the original image and calculating the coefficient vector c for the linear predictive model. An index variable, i , is initialized to 1. The core of the algorithm operates within a main loop that iterates for every pixel. Within this loop:

- The algorithm calculates the intermediate predicted pixel value ($\hat{a}^i$) using the coefficients and neighborhood pixel values.
- The adjusted predicted value ($\hat{a}^i$) is obtained by forcing ($\hat{a}^i$) to be an even integer.
- The prediction error (p^i) is then calculated as the difference between the actual pixel value (a^i) and the adjusted predicted value ($\hat{a}^i$).

The flow then checks the absolute value of the prediction error (p^i) against a predefined threshold (T). This core logic determines the modifiability of the pixel.

For errors within the threshold ($|p^i| \leq T$), the algorithm checks for expandability. If the error is expandable, it is expanded ($p^i = 2p^i$), and the local map indicator L is set to 1. If it is not expandable, the local map indicator L is set to 0, and the necessary LSB is collected.

For errors outside the threshold ($|p^i| > T$), the algorithm checks for shiftability. If the error is shiftable, it is shifted ($p^i = p^i + \text{sign}(p^i)(T + 1)$), and L is set to 1. If it is not shiftable, L is set to 0, and the required LSB is collected.

After the error adjustment logic, the original pixel value ($\hat{a}^i$) is recovered using the adjusted prediction error (p^i), and the loop index is incremented.

Once the main iteration concludes, the collected information is compiled into the final BITSTREAM. A second loop commences to perform the final embedding:

- The algorithm embeds the watermark bit (w) into the adjusted prediction error (p^i) to form the watermarked error (p_w^i).
- The final watermarked pixel (a_w^i) is generated by adding the watermarked prediction error (p_w^i) back to the predicted pixel value ($\hat{a}^i$).
- The loop index is incremented and the process stops once all pixels have been processed.

4. Data Extraction and Image Recovery. We first decode an embedded bit stream from all pixels LSB, $w = \text{mod}(p_w, 2)$ and then separate image feature information from watermark. Since the context of a pixel is required to be restored before the pixel itself can be restored, the restoration algorithm is sequential. For each pixel is arranged horizontally from left to right and $\hat{p}$ is a watermarked prediction-error after extracting its LSB, a restoration or watermark extracting procedures can be summarized and represented by the pseudo code as shown in Figure 8, while the algorithm of the proposed watermark extracting can be visualized by the flowchart diagram shown in Figure 9.

The procedures for watermark extraction and image recovery are detailed in the pseudo code shown in Figure 8. This algorithm outlines a five-step process that systematically reverses the embedding operation, extracting the hidden data while achieving the lossless recovery of the original image pixels. The process begins with Step 1, where the predicted pixel value ($\hat{a}$) is calculated from the neighboring pixels of the watermarked image (a_w), utilizing the same linear predictive model employed during the embedding phase. Step 2 calculates the prediction error components from the watermarked image. Initially, the algorithm computes the watermarked prediction error (p_w) by finding the difference between the watermarked pixel (a_w) and the predicted value ($\hat{a}$). Subsequently, the algorithm determines the extracted prediction error component ($\hat{p}$) using the floor function operation on (p_w). The crucial part of the recovery process occurs in Step 3, which focuses on recovering the original prediction error (p). This step involves checking whether the

current pixel is a member of the predefined sets S_1 or S_2 , using the criteria established by Equations (12) and (13). The logic is structured as a series of checks.

- If the pixel satisfies the S_1 condition, the original prediction error (p) is recovered using Equation (6).
- If the S_1 condition is not met (Else), the algorithm checks if the pixel satisfies the S_2 condition; if so, (p) is recovered using Equation (9).

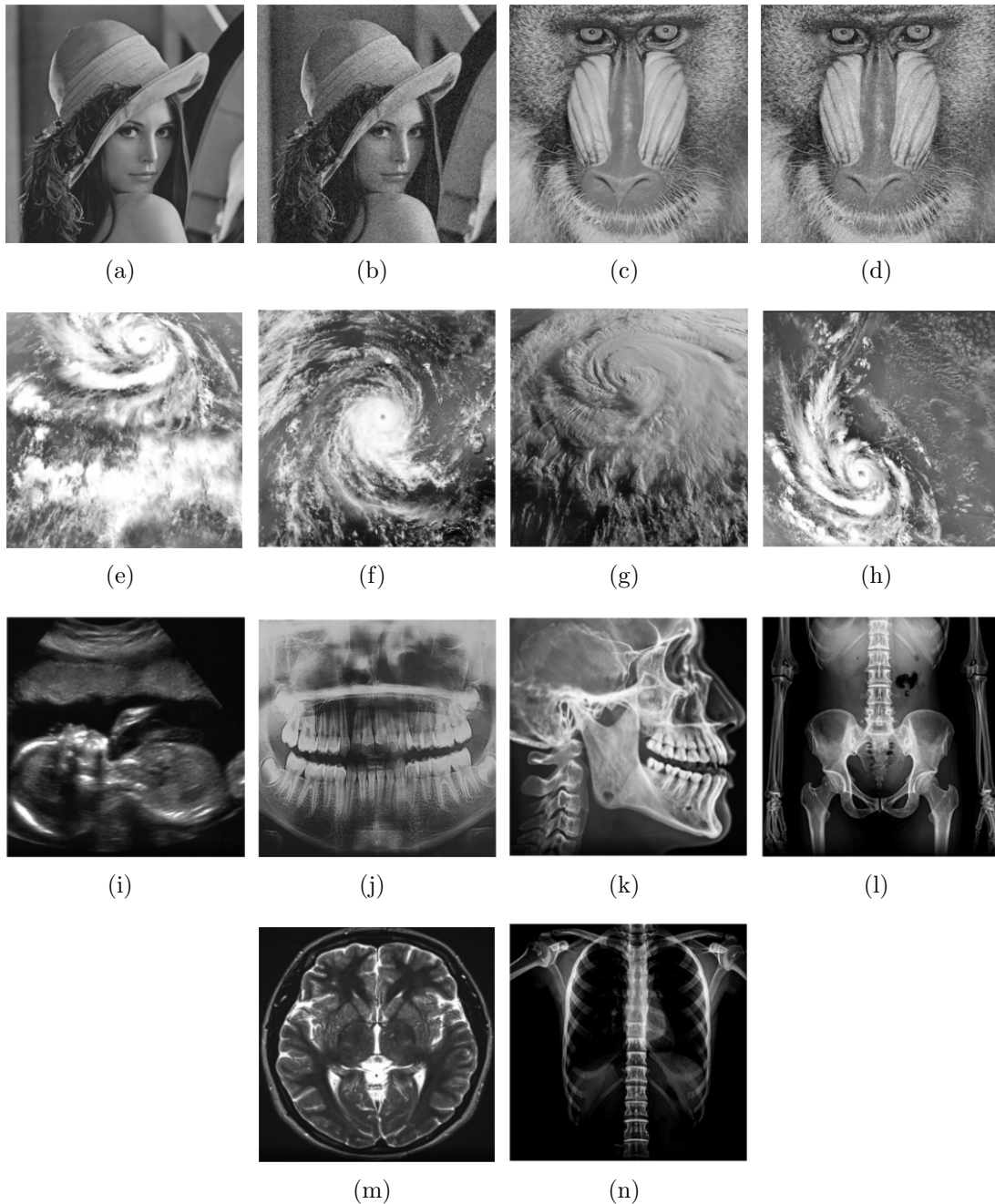
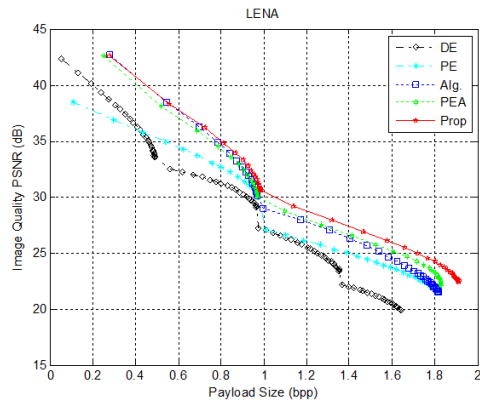
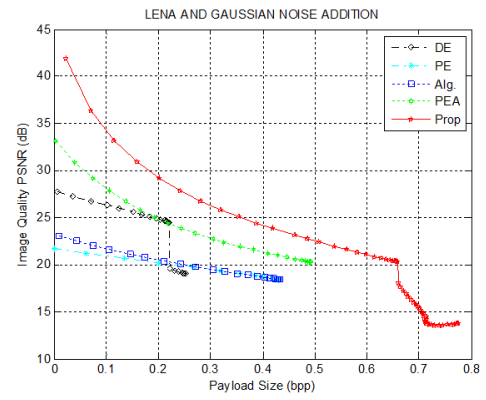


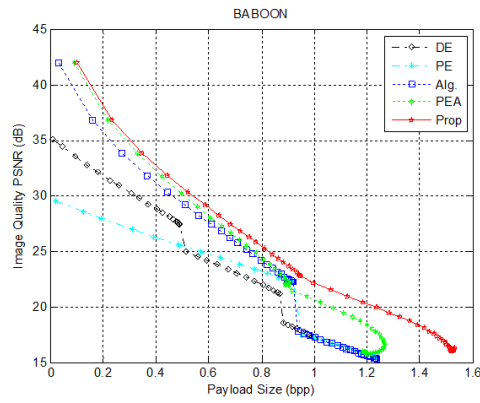
FIGURE 6. 512×512 grayscale test images and some specific images: (a) Lena; (b) Lena and Gaussian noise addition; (c) Baboon; (d) Baboon and Gaussian noise addition; (e)-(h) satellite image 1-4; (i)-(n) medical image 1-6



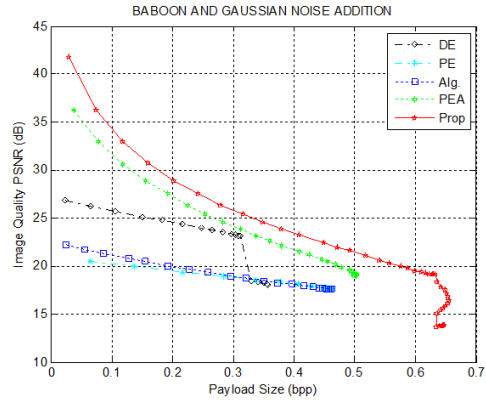
(a)



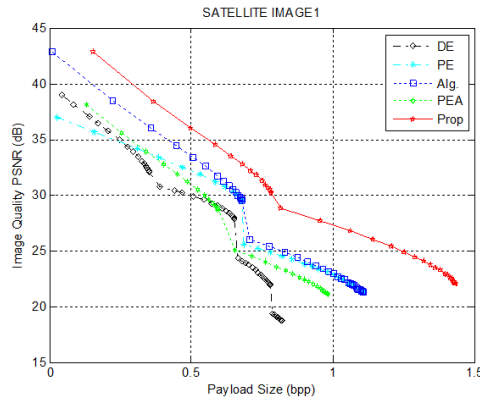
(b)



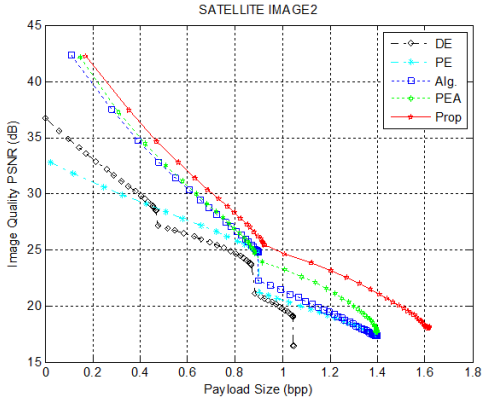
(c)



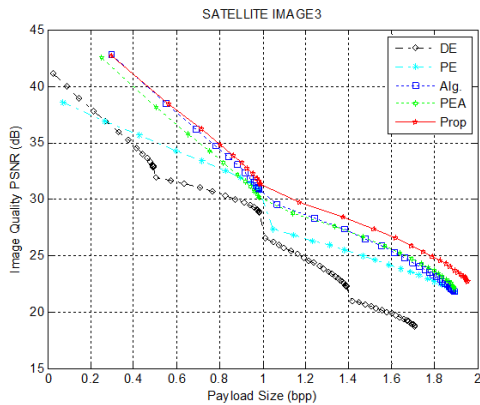
(d)



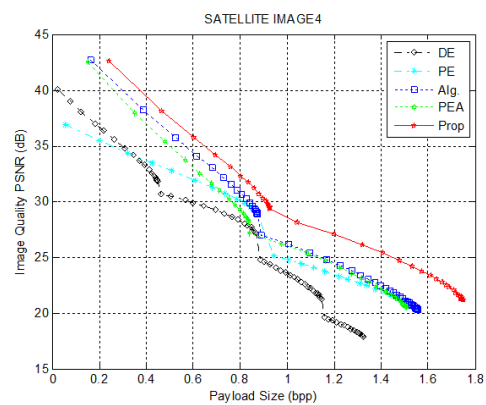
(e)



(f)



(g)



(h)

(Continued)

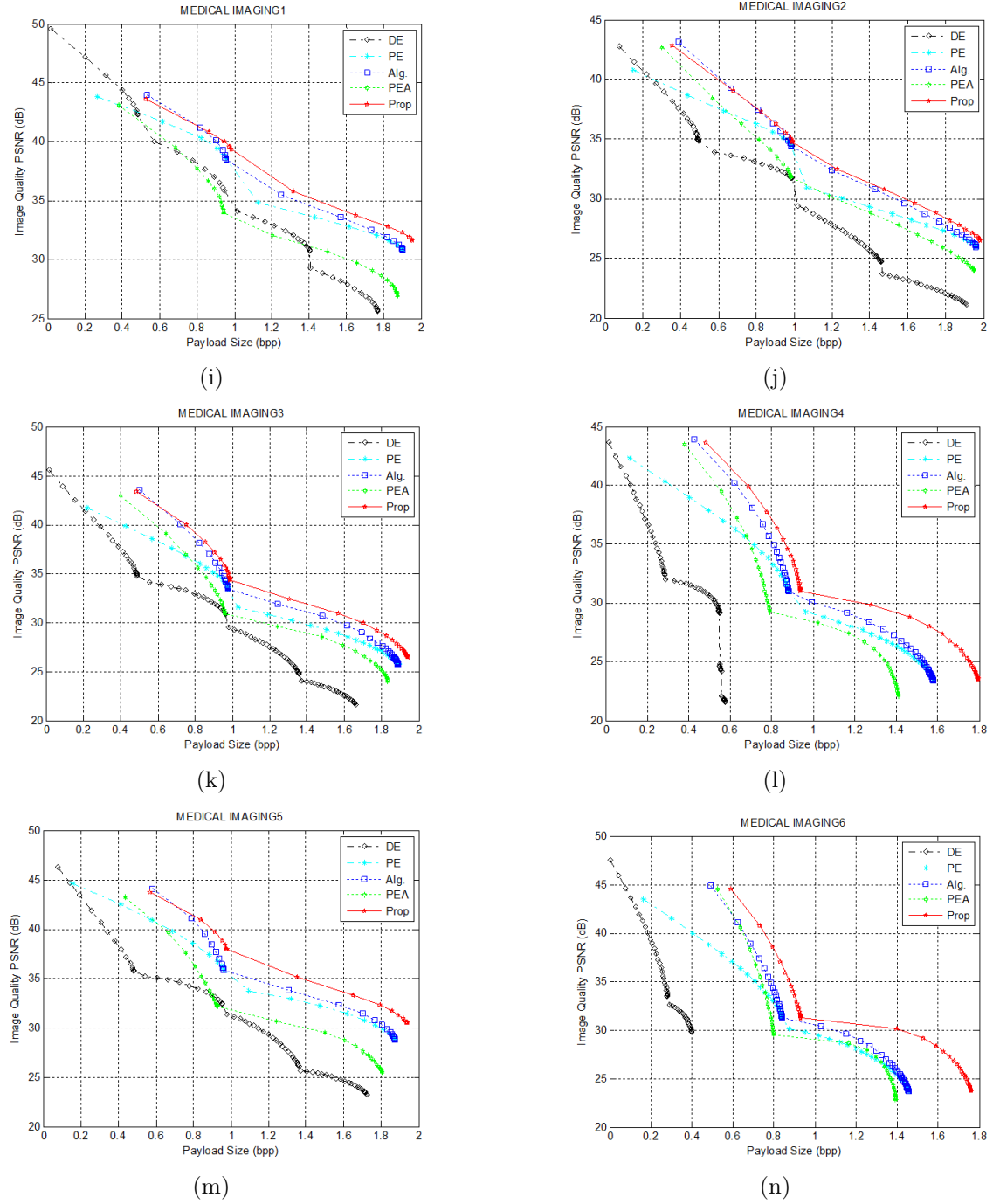


FIGURE 7. Performance of testing images in Figure 6: (a) Lena; (b) Lena and Gaussian noise addition; (c) Baboon; (d) Baboon and Gaussian noise addition; (e)-(h) satellite image 1-4; (i)-(n) medical image 1-6

- If neither the S_1 nor S_2 condition is met (Else), the algorithm proceeds to find the pixel's recovery formula from the rest of the Prediction-Error Adjustment (PEA) map, ensuring that all adjusted errors can be restored.

Once the original prediction error (p) is successfully recovered, Step 4 recovers the original pixel value (a). This is achieved by adding the recovered original prediction error (p) back to the predicted pixel value ($\hat{a}$), as expressed by the formula $a = \hat{a} + p$. Finally, the entire procedure is defined as recursive. Step 5 requires the algorithm to repeat the entire process

Watermark Extracting procedures:

Step 1: Calculate the predicted pixel value $\hat{a}$ from neighborhood pixels

Step 2: Calculate the prediction error components: $\hat{p}$, $p_w = a_w - \hat{a}$ and $\hat{p} = 2\lfloor p_w/2 \rfloor$

Step 3: Check if a pixel is a member of S_1 or S_2 using (12) and (13).
 If the pixel satisfies S_1 condition, recover p using Equation (6)
 else if the pixel satisfies S_2 condition, recover p using Equation (9)
 else finding pixel's recovery formula from the rest of PEA map.

Step 4: Recover the original pixel value: a , $a = \hat{a} + p$

Step 5: Repeat the procedure recursively until all original pixel values are restored

FIGURE 8. Pseudo code of the watermark extracting procedures

recursively until all pixels in the watermarked image are fully processed and all original pixel values are completely restored, thus completing the reversible data hiding cycle.

Figure 9 illustrates the comprehensive flowchart for the proposed reversible watermark extracting algorithm. The process begins by reading the watermarked image and extracting the initial bitstream, which allows for the recovery of the map, LSBs, and watermark data. The main loop iterates through each pixel, where the algorithm calculates the predicted pixel value and the watermarked prediction error. A core check determines if the error is within the shifting threshold. Based on whether the error is expandable or shiftable, the algorithm recovers the hidden watermark bit, loads the necessary LSB from the recovered bitstream, and restores the original prediction error. The local map indicator guides the process of loading LSBs. Finally, the original pixel value is recovered by adding the restored prediction error back to the predicted value, and the process repeats until the entire original image is fully restored.

5. Simulation and Experimental Results. We implemented the proposed research using MATLAB R2014a as the primary simulation environment. A dataset comprising various 512×512 grayscale test images and several specific images was employed. Figure 6 illustrates a selection of these images, including standard benchmarks like (a) Lena and (c) Baboon, their noisy counterparts (b) Lena and (d) Baboon with Gaussian noise (zero mean, 0.009 SD), (e)-(h) four satellite images, and (i)-(n) six medical images. For the lossless compression of the Prediction-Error Adjustment (PEA) map, the algorithms utilized an arithmetic encoder. The proposed algorithm, reduced PEA with linear predictive function (Prop), was systematically compared against four established former algorithms: Different Expansion (DE), Prediction-Error Expansion (PE), PE with histogram shifting and overflow map (Alg.), and PE Adjustment (PEA). All figures show the results obtained after embedding a double-layer watermark into the test images. The results demonstrate the effectiveness of the proposed algorithms. Specifically, the linear predictive function performs remarkably well with images exhibiting high pixel value variation. Figure 7 illustrates that the Prop method achieved a superior result for images with added Gaussian noise, such as (b) Lena and (d) Baboon, compared to the original images without noise, (a) Lena and (c) Baboon. Furthermore, the method was tested on application-specific images, including satellite and medical images, which naturally contain noise. Figures 7(e)-7(n) confirm that the linear predictive function is well-suited for this category of noisy images. A crucial advantage stems from the reduced PEA map size, which allows the watermark to occupy more space, consequently increasing the maximum information

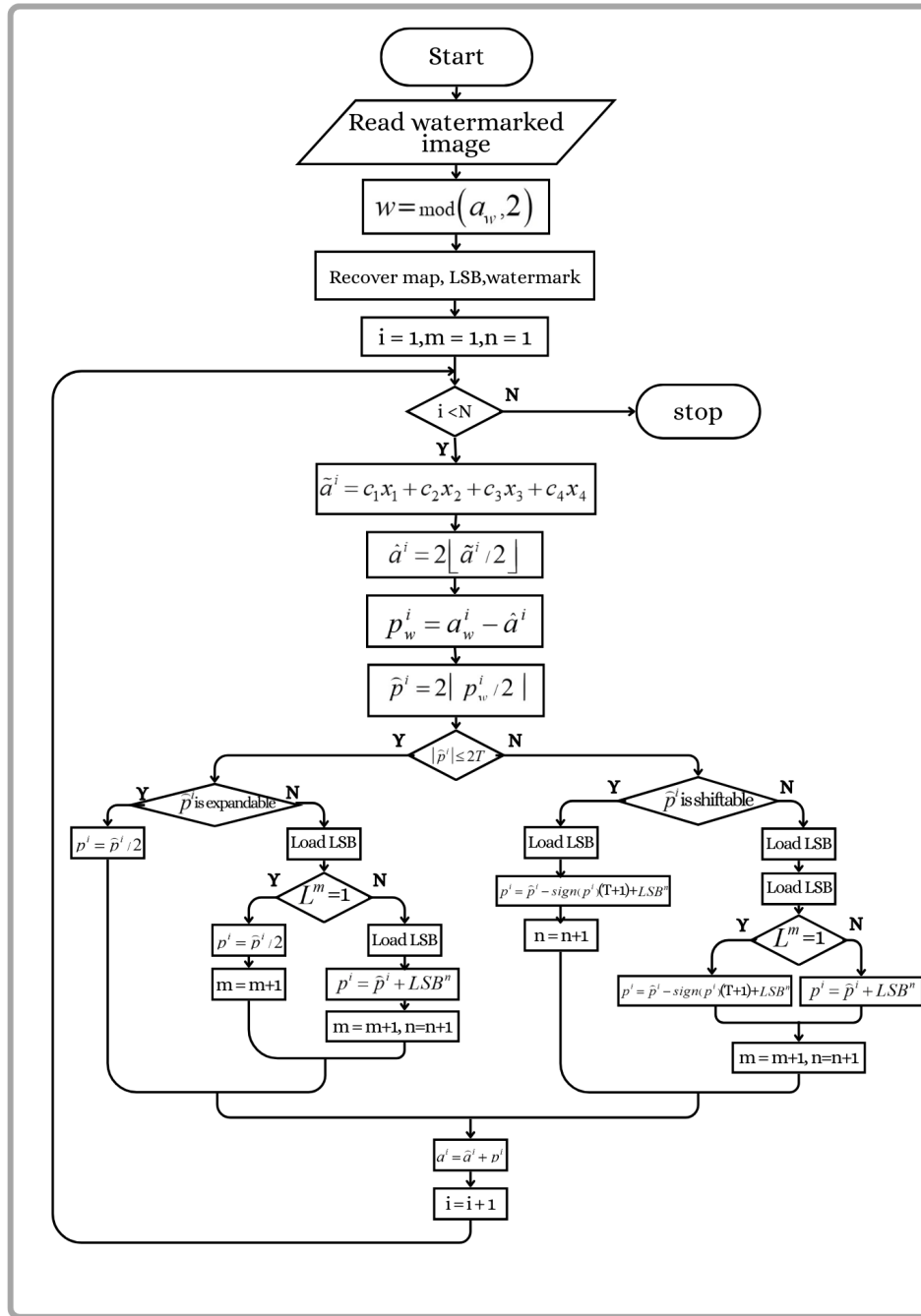


FIGURE 9. Flowchart of the proposed watermark extracting algorithm

hiding capacity for each image. When considering stepped-down graphs due to the multiple layers of watermarks, the linear predictive function performs better than the other predictors. This performance creates a wider gap between the Prop method and the former techniques. Significantly, Prop yields the highest PSNR for the same payload size, particularly evident in the second embedding layer. This superiority occurs because the embedded image contains greater pixel value variation, which inherently decreases the effectiveness of other predicting functions, while the linear predictive function maintains its strong performance. Finally, Figures 7(a)-7(n) reveal that Prop offers a significantly higher watermarking capacity compared to the established methods, whose maximum capacity limit is substantially lower.

Table 1 summarizes the comparative performance of the proposed APEC method and four benchmark reversible watermarking techniques (DE, PE, Alg., and PEA) across the fourteen test images shown in Figures 6(a)-6(n). In addition to embedding capacity and visual quality, the comparison also considers computational complexity and average runtime to provide a clearer view of the practical trade-offs associated with each method. From a computational perspective, the classical DE, PE, Alg., and PEA methods all exhibit linear complexity with respect to the number of pixels, i.e., $O(N)$, since their operations primarily involve pixel-wise prediction-error expansion and histogram shifting. The proposed APEC method introduces additional processing steps, including constructing of a reduced prediction-error adjustment map and the estimation of linear prediction coefficients using least-squares optimization. Consequently, the theoretical complexity becomes $O(Nk^3)$, where k denotes the number of predictor variables. In practice, however, k is small and fixed (four neighboring pixels), and the coefficient estimation is performed only once per image. Therefore, the additional computational overhead remains limited. The runtime measurements reported in Table 1 confirm this observation. While the proposed method requires a slightly higher execution time (0.92 s on average) compared with the simpler baseline approaches, the increase remains moderate relative to the overall watermark embedding procedure. This overhead is compensated by significant improvements in embedding performance, such as increased data capacity and enhanced robustness against attacks.

TABLE 1. Comparative performance evaluation of the proposed APEC method against benchmark reversible watermarking techniques (DE, PE, Alg., PEA) across 14 test images (Figures 6(a)-6(n))

Method	Complexity (Big O)	Avg. runtime (s)	Avg. embedding capacity (bpp)	Avg. Max PSNR (dB)	Prediction strategy	Practical trade-off analysis
DE	$O(N)$	0.23 ± 0.00	0.40 ± 0.00	29.19 ± 7.83	Pixel difference	fastest; but lowest capacity and quality, especially for medical images
PE	$O(N)$	0.37 ± 0.00	1.25 ± 0.00	31.50 ± 6.40	Fixed predictor	low complexity; prone to high prediction errors in textures regions
Alg	$O(N)$	0.42 ± 0.00	1.45 ± 0.00	32.61 ± 6.24	Median edge detection	moderate performance; limited adaptive capability.
PEA	$O(N)$	0.57 ± 0.00	1.65 ± 0.00	33.98 ± 5.76	Simple adaptive predictor	balanced performance; but location map overhead remains significant.
APEC	$O(Nk^3)$	$0.92 \pm 0.09^*$	$1.85 \pm 0.00^*$	$37.84 \pm 4.88^*$	Least-squares optimization	higher capacity and PSNR; moderate computational overhead due to predictor estimation

*Statistically significant difference compared with the best-performing benchmark (PEA), ($p < 0.05$).

In particular, the proposed APEC method achieves the highest average embedding capacity (1.85 bpp) and the highest average maximum PSNR (37.84 dB) among all compared techniques. These improvements arise from two main factors. First, the reduced APEC

map decreases the auxiliary data overhead, allowing a larger portion of the image to be used for watermark embedding. Second, the least-squares based linear predictor provides more accurate prediction of pixel values, thereby reducing prediction errors and increasing the number of expandable pixels. These findings demonstrate that APEC provides an attractive trade-off for practical applications. Although the method requires slightly higher computational effort than the fixed-step benchmark approaches, the additional cost is offset by substantial improvements in embedding capacity and visual quality. Consequently, the proposed framework is well suited for applications requiring both high payload and strict lossless fidelity, such as medical imaging and satellite remote sensing.

6. Conclusions and Future Work. This paper introduces a novel linear predictive modeling approach, specifically for optimizing the design of the linear predictive function used in digital watermark embedding. We also developed the reduced APEC map as an enhancement to the existing PEA map reduction method. The algorithms proposed in this study demonstrate significant effectiveness, yielding high-quality image watermarking alongside high embedding capacity. A key contribution lies in the successful reduction of the PEA map size, which provides more embedding space while maintaining the target PSNR (Peak Signal-to-Noise Ratio). Furthermore, our comparative analysis showed that the linear predictive function outperforms other functions when applied to images characterized by high pixel value variation. This finding establishes the method as particularly suitable for specialized applications where input images inherently contain high levels of noise, such as X-ray or satellite imagery. The robustness and superior performance in these conditions represent a notable advancement in the field. The promising results of this study encourage further refinement of the proposed approach. Future work will investigate alternative prediction models to improve prediction accuracy, particularly for images with low pixel-value variation where linear predictors may be inadequate. Potential directions include the use of nonlinear prediction models, such as polynomial or kernel-based regression, as well as machine learning based predictors (e.g., decision tree regression, support vector regression, and shallow neural networks) to better capture complex spatial relationships among neighboring pixels. Improved prediction accuracy is expected to reduce prediction errors and increase the number of expandable pixels, thereby enhancing both embedding capacity and visual quality. These efforts aim to further extend the applicability and performance of the proposed reversible watermarking scheme.

REFERENCES

- [1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich and T. Kalker, *Digital Watermarking and Steganography*, 2nd Edition, Morgan Kaufmann Publishers, Burlington, 2008.
- [2] H.-Y. Fan, Z.-M. Lu and Y. Liu, The digital image watermarking scheme using low frequency construction and histogram, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.367-384, 2020.
- [3] W. Meng, Q. Zheng and F. Wang, Research on watermarking algorithm based on improved Arnold permutation and adaptive embedding position, *International Journal of Innovative Computing, Information and Control*, vol.19, no.6, pp.1847-1859, 2023.
- [4] X. Zhong, A. Das, F. Alrasheedi and A. Tanvir, A brief, in-depth survey of deep learning based image watermarking, *Appl. Sci.*, vol.13, no.11, 11852, 2023.
- [5] R. Liang and F. Li, Design and research of a multi-view graph deep learning 3D model retrieval system based on fusion Vision-Transformer, *International Journal of Innovative Computing, Information and Control*, vol.20, no.6, pp.1775-1788, 2024.
- [6] Z. Li, Dynamic network representation learning method combining graph neural networks and temporal information, *International Journal of Innovative Computing, Information and Control*, vol.20, no.6, pp.1803-1817, 2024.

- [7] S. G. Jachak, J. Mahajan, T. Kshirsagar, S. Gaikwad and P. Mule, Digital watermarking and steganography, *Int. Res. J. Adv. Eng. Manag.*, vol.3, no.3, pp.969-976, 2025.
- [8] W. Bender, D. Gruhl, N. Morimoto and A. Lu, Technique for data hiding, *IBM Syst. Journal*, vol.35, no.3, pp.313-336, 1996.
- [9] J. M. Barton, *Method and Apparatus for Embedding Authentication Information within Digital Data*, U.S. Patent 5 646 997, 1997.
- [10] B. Macq, Lossless multiresolution transform for image authenticating watermarking, *2000 10th European Signal Processing Conference*, Tampere, Finland, pp.533-536, 2000.
- [11] C. W. Honsinger, P. Jones, M. Rabbani and J. C. Stoffel, *Lossless Recovery of an Original Image Containing Embedded Data*, U.S. Patent 6 278 791, 2001.
- [12] C. De Vleeschouwer, J. E. Delaigle and B. Marq, Circular interpretation of bijective transformation in lossless watermarking for media asset management, *IEEE Trans. Multimedia*, vol.5, no.1, pp.97-105, 2003.
- [13] J. Fridrich, M. Goljan and R. Du, Lossless data embedding – New paradigm in digital watermarking, *EURASIP Journal Appl. Signal Process.*, vol.2, pp.185-196, 2003.
- [14] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su, Distortionless data hiding based on integer wavelet transform, *IEE Electron. Lett.*, vol.38, no.25, pp.1646-1648, 2002.
- [15] J. Tian, Reversible watermarking using a difference expansion, *IEEE Trans. Circuits Syst. Video Technol.*, vol.13, no.8, pp.890-896, 2004.
- [16] 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, 2005.
- [17] D. M. Thodi and J. J. Rodriguez, Prediction-error based reversible watermarking, *Proc. of IEEE Conf. Image Processing*, pp.1549-1552, 2004.
- [18] D. M. Thodi and J. J. Rodriguez, Expansion embedding techniques for reversible watermarking, *IEEE Trans. Image Process.*, vol.16, no.3, pp.721-730, 2007.
- [19] S. Weng, Y. Zhao, R. Ni and J.-S. Pan, Lossless data hiding based on prediction-error adjustment, *Sci. China Ser. F-Inf. Sci.*, vol.52, no.2, pp.269-275, 2009.
- [20] S.-C. Huang, A reversible visible watermarking technique for BTC-compressed images, *International Journal of Innovative Computing, Information and Control*, vol.15, no.6, pp.2211-2227, 2019.
- [21] X. Tang, L. Zhou, D. Liu, B. Liu and X. Li, Reversible data hiding based on improved rhombus predictor and prediction error expansion, *Proc. of 2020 IEEE 19th Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp.1234-1241, 2020.
- [22] L.-C. Huang, S.-F. Chiou and M.-S. Hwang, A reversible data hiding based on histogram shifting of prediction errors for two-tier medical images, *Informatika*, vol.32, no.1, pp.69-84, 2021.
- [23] A. K. Rai, H. Om, S. Chand and C.-C. Lin, High-capacity reversible data hiding based on two-layer embedding scheme for encrypted image using blockchain, *Computers*, vol.12, no.6, 120, 2023.
- [24] S. Panchikkil, S. P. Vegesana, V. M. Manikandan, P. K. Donta, P. K. R. Maddikunta and T. R. Gadekallu, An ensemble learning approach for reversible data hiding in encrypted images with Fibonacci transform, *Electronics*, vol.12, no.2, 450, 2023.
- [25] M. A. Tahiri, H. Karmouni, M. Sayyouri, H. Qjidaa, M. Ahmad, M. Hammad, P. Pławiak, O. Alfarraj and A. A. Abd El Latif, An improved reversible watermarking scheme using embedding optimization and quaternion moments, *Sci. Rep.*, vol.14, 18485, 2024.
- [26] Y. Qiu, W. Peng and X. Lin, Improved CNN prediction based reversible data hiding for images, *Entropy*, vol.27, no.2, 159, 2025.
- [27] H. Ragab, H. Shaban, K. Ahmed and A. Ali, Digital image steganography and reversible data hiding: Algorithms, applications and recommendations, *J. Image Graphics*, vol.13, no.1, pp.90-114, 2025.
- [28] J. Chen, W. Wang, C. Shi, L. Dong, Y. Li and X. Hu, Deep robust reversible watermarking, *arXiv Preprint*, arXiv: 2503.02490, 2025.
- [29] W.-Y. Yang, W. Cao, T. S. Chung and J. Morris, *Applied Numerical Methods Using MATLAB*, John Wiley and Sons, New Jersey, 2005.

Author Biography



Thanet Ketthong received his B.Eng. degree in Electronics and Telecommunication Engineering from Pathumwan Institute of Technology (PIT), Thailand, in 2000. He then earned his M.Eng. degree in Electrical Engineering from King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 2011. He completed his doctoral studies, receiving the D.Eng. degree in Electrical Engineering from PIT in 2017.

Dr. Ketthong is currently a full-time associate professor in the Department of Electrical Engineering at Dhonburi Rajabhat University, Thailand. His diverse research interests include system identification, intelligent control, power electronics, and motor drives. Furthermore, he specializes in image processing, biomedical and digital signal processing, and the application of metaheuristic algorithms to various real-world industrial and electrical engineering problems. He has published his work in numerous journals and conference proceedings.



Lucksana Romyasamit received her B.B.A. degree in Business Computer from Siam University, Thailand, in 1993. She then earned her M.Sc. degree in Science and Technology Education and her Ph.D. degree in Computer Education from King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2008 and 2021, respectively.

Dr. Romyasamit is currently a lecturer in the Department of Computer Science, Faculty of Science and Technology, at Dhonburi Rajabhat University, Thailand. Her primary research focus is the integration of participatory technology with communities.