

Deep Prediction and Efficient 3D Mapping of Color Images for Reversible Data Hiding

Runwen Hu^{ID}, Yuhong Wu, Shijun Xiang^{ID}, Xiaolong Li^{ID}, Member, IEEE, and Yao Zhao^{ID}, Fellow, IEEE

Abstract—In the reversible data hiding (RDH) community, both prediction and mapping strategies are vital for reducing distortion. With high prediction performance, small prediction errors can be generated to reduce the embedding distortion. Besides, the efficient mapping strategy can improve the practicality. In this paper, we propose a new RDH method for color images by using convolution neural networks (CNNs) for prediction and an efficient 3D mapping strategy for embedding. At first, each color image is elaborately divided into three isolated image sets so that the proposed deep prediction network (DPN) can exploit more neighboring pixels in the current channel and the correlation between three channels. Then, an efficient 3D mapping strategy is luminously designed by using the symmetry of the 3D prediction error histogram (PEH). The symmetry of 3D PEH has been analyzed in statistical and experimental ways. Based on the proposed deep prediction network and efficient 3D mapping strategy (DPEM), we construct an efficient RDH method for color images. The performance of the proposed DPN is evaluated by comparing it with several predictors on different image datasets. The embedding performance has been demonstrated by hiding information in color images, e.g., the average PSNR value of the Kodak dataset is 63.63 dB with an embedding capacity of 50,000 bits. Furthermore, the experimental results on the ImageNet and PASCAL VOC2012 datasets have shown the proposed RDH method is superior to several state-of-the-art RDH methods. With the introduction of deep learning, the development of the RDH method for color images can be promoted.

Index Terms—Reversible data hiding, convolutional neural network, efficient 3D mapping, color images.

I. INTRODUCTION

REVERSIBLE data hiding (RDH) has demonstrated its utility in addressing various multimedia security challenges [1], [2]. Being able to extract the hidden information correctly and recover the original contents losslessly, RDH

Received 12 August 2024; revised 5 December 2024 and 4 February 2025; accepted 11 February 2025. Date of publication 24 February 2025; date of current version 5 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272197, Grant 62372037, and Grant U24B20179. The associate editor coordinating the review of this article and approving it for publication was Dr. Roberto Caldelli. (*Corresponding author: Shijun Xiang*.)

Runwen Hu is with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: runwen_hu@qq.com).

Yuhong Wu and Shijun Xiang are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: yuhong_wu@qq.com; Shijun_Xiang@qq.com).

Xiaolong Li and Yao Zhao are with the Institute of Information Science and the Visual Intelligence + X International Cooperation Joint Laboratory of MOE, Beijing Jiaotong University, Beijing 100044, China (e-mail: lixl@bjtu.edu.cn; yzhao@bjtu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2025.3544956

is extensively employed in the integrity authentication of multimedia, including images, audio, and videos [3]. For sensitive fields such as medicine and the military, RDH has many practical applications and is a promising direction.

In the RDH community, prediction and embedding techniques are two primary research topics. The prediction techniques aim to reduce prediction errors, while the embedding techniques focus on minimizing the embedding distortion. By integrating these two techniques, effective RDH methods can be developed. For image-related works, although significant advancements have been made in RDH methods for grayscale images, research on color images remains limited. Given the widespread use of color images, further exploration of RDH methods for these images is highly valuable.

In the prediction aspect, several predictors have been proposed for grayscale images, including the difference predictor (DP) [4], [5], median edge detection predictor (MEDP) [6], gradient adaptive predictor (GAP) [7], rhombus predictor (RP) [8], multi-predictors [9], etc. For color images, research on prediction techniques is mainly based on the RP. For instance, raw color images have been predicted using the RP [10]. Li et al. [11] utilized the RP along with edge information between three channels for joint prediction. Based on the RP and least square prediction method [12], a double-layer least square prediction method is proposed [13]. Yao et al. [14] proposed a guided filtering predictor to leverage the correlation for predicting the pixels. Based on the work in [14], Peng et al. [15] developed a new RDH method for color images by using the multi-channel synchronized histogram. Additionally, Mao et al. [16] utilized the channel correlation to propose an efficient RDH method for color images. Among these prediction methods, most methods only consider a few neighboring pixels in the current channel or simple correlations between the three channels. To improve the accuracy, more neighboring pixels and the correlation between three channels of color images should be considered.

Compared with the classical predictors, using deep learning as the prediction technique in the RDH community is a new trend. Based on the capabilities of feature extraction and multi-receptive fields, convolution neural networks (CNNs) are suitable for image processing tasks [17], [18]. In the field of RDH, CNN-based predictor has been used for stereo images [19] and grayscale images [20], [21], [22]. Though previous CNN-based predictors utilize more neighboring pixels, the correlation between three channels of color images is not considered. To this end, we propose a new deep prediction

network (DPN) for color images by using CNN, which can utilize more neighboring pixels in the current channels and exploit the correlation between three channels.

In the embedding aspect, prediction errors, calculated from the original images and the predicted images, are usually used for embedding. Several embedding techniques for grayscale images have been proposed, including lossless compression [23], [24], difference expansion (DE) [4], [25], prediction error expansion (PEE) [6], histogram shifting (HS) [2], [26], [27], prediction error histogram (PEH) [8], [28], [29], [30], pixel-value-ordering (PVO) [31], [32], etc. For color images, previous methods focus on embedding information independently into each channel of color images [14], [33], [34], [35]. Furthermore, techniques such as weighted matrix [36], [37], [38], [39] and the coding methods [40], [41], [42], [43] are applied to improve the embedding performance. To utilize the correlation, Hou et al. considered the chromaticity space conversion to keep the grayscale image unchanged [44]. Zhan et al. [45] introduced a fixed three-dimensional (3D) mapping strategy. Furthermore, Chang et al. proposed an efficient 3D mapping selection mechanism [46]. In [47], the double deep Q-network [48] and reinforcement learning are adopted to find 3D mappings. Despite the advancements, 3D mapping-based methods often involve high computational complexity and time-consuming generation processes. To address this, we propose an efficient 3D mapping strategy that exploits the symmetry of 3D PEH, making it suitable for embedding binary information.

In the research of RDH methods, the visual quality of embedded images might decrease if the prediction performance is inadequate, and the applicability of RDH method may be constrained if the embedding strategy is time-consuming. To address these challenges, we propose a new RDH method for color images by innovating both the prediction and the embedding techniques. In the prediction part, a new image division method is designed to divide the cover images, providing two-thirds of the image information as the prediction contexts. Subsequently, the DPN is proposed to improve the prediction performance by using more neighboring pixels and the correlation between three color channels. In the embedding part, we investigate the symmetry of 3D PEH to propose an efficient 3D mapping strategy by simplifying the design of 3D mappings into 2D mappings. By integrating the deep prediction network and efficient 3D mapping strategy (DPEM), the proposed RDH method achieves superior performance, surpassing several state-of-the-art RDH methods for color images [14], [34], [45], [46], [47].

The rest of this paper is organized as follows. Section II describes the structure of the proposed RDH method. Section III and Section IV illustrate the prediction part and the embedding part of the proposed RDH method, respectively. In Section V, the experimental results and comparisons are provided. Finally, we conclude this paper in Section VI.

II. STRUCTURE OF THE PROPOSED RDH METHOD

In this section, we introduce the structure of the proposed RDH method for color images, as shown in Fig. 1. The

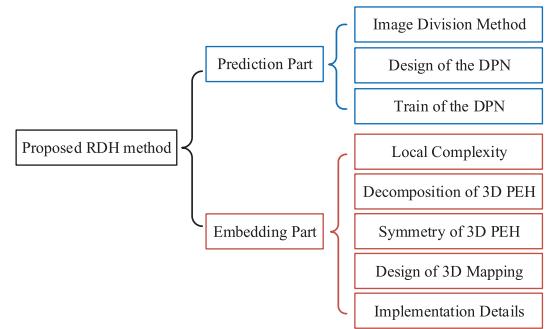


Fig. 1. The structure of the proposed RDH method.

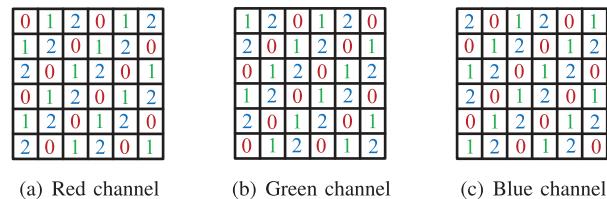


Fig. 2. Illustration of dividing three channels of a color image into three independent parts: (a) red channel, (b) green channel, and (c) blue channel.

proposed DPN is designed in Section III, while the efficient 3D mapping strategy is proposed in Section IV.

In the prediction part, a new color image division method is proposed in Section III-A to divide the color images, providing two-thirds of the image information as the prediction contexts. After that, Section III-B describes the proposed DPN, which can utilize more neighboring pixels and exploit the correlation between three channels. Finally, the training details are provided in Section III-C.

In the embedding part, the local complexity calculation method is first proposed to select the prediction error triplet (PET) with small absolute values in Section IV-A. After that, the 3D PEH is decomposed to discuss the symmetry in Section IV-B and Section IV-C. Based on the symmetry, Section IV-D proposes an efficient 3D mapping strategy. Finally, the implementation details are described in Section IV-E.

III. PREDICTION PART

To improve the prediction performance, we first propose a new image division method to provide more information of color images. Following this, the design of the DPN is illustrated. Finally, the training details are described.

A. Image Division Method

In the RDH community, it is important to provide extensive image information as the prediction context for CNN-based methods to improve the prediction performance [19], [20], [21], [22]. When designing the image division method for color images, the correlation between the three channels is fully considered.

Assuming that the color image $I \in \mathbb{R}^{M \times N \times 3}$ is represented in RGB space, the pixels at position (i, j) in the red, green, and

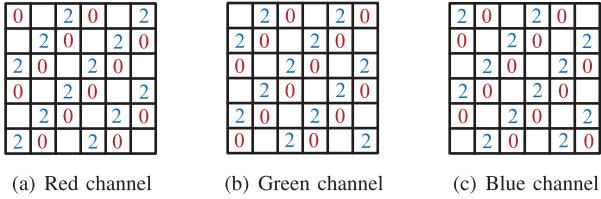


Fig. 3. Illustration of the image I_{02} : (a) red channel, (b) green channel, and (c) blue channel.

blue channels are labeled as $x_{i,j}^r$, $x_{i,j}^g$, and $x_{i,j}^b$, respectively. In general, the pixels with the same locations in three channels have similar values. To utilize this characteristic, we propose a new color image division method. Fig. 2 illustrates this division method with a 6×6 sized block, in which Fig. 2 (a), (b), and (c) show the division patterns in red, green, and blue channels, respectively. Since each pixel is marked with one of the label “0”, “1”, and “2”, three sets of images $I_0 \in \mathbb{R}^{M \times N \times 3}$, $I_1 \in \mathbb{R}^{M \times N \times 3}$, and $I_2 \in \mathbb{R}^{M \times N \times 3}$ can be defined. Based on this image division method, the proposed RDH method is processed on the divided images I_0 , I_1 , and I_2 independently.

Take the pixels in image I_1 for further description. The pixels at the same locations in other channels belong to images I_0 or I_2 . To predict the pixels in image I_1 using the proposed DPN, the pixels in images I_0 and I_2 are combined to build the input image $I_{02} \in \mathbb{R}^{M \times N \times 3}$, which provides two-thirds of the image information as the prediction contexts. Fig. 3 illustrates image I_{02} using a 6×6 sized block, in which the red, green, and blue channels are shown in Fig. 3 (a), (b), and (c), respectively. The image I_{02} may become sparse matrices if the blank blocks are filled with value 0. On one hand, the sparse matrices may affect the stability of CNN-based methods [49], [50]. On the other hand, the dense matrices may provide better performance [51]. In this light, the blank blocks in image I_{02} are filled with interpolated values to create dense matrices. For the to-be-predicted pixel $x_{i,j}$, the interpolated value $\tilde{x}_{i,j}$ is calculated as follows:

$$\tilde{x}_{i,j} = (x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1})/4, \quad (1)$$

where $x_{i-1,j}$, $x_{i+1,j}$, $x_{i,j-1}$, and $x_{i,j+1}$ are four neighboring pixels in the same channel as $x_{i,j}$. Similarly, to predict the images I_0 and I_2 , different input images I_{21} and I_{10} should be constructed. Specifically, the input image I_{21} ensures that the target image I_0 is positioned in the middle of the three channels, maintaining consistency in the image prediction process. Similar operations are applied to construct the input image I_{10} for the target image I_2 . In this way, the prediction process can be formulated as:

$$\begin{cases} \hat{I}_0 = \text{DPN}(I_{21}) \\ \hat{I}_1 = \text{DPN}(I_{02}) \\ \hat{I}_2 = \text{DPN}(I_{10}), \end{cases} \quad (2)$$

where \hat{I}_0 , \hat{I}_1 , and \hat{I}_2 are the predicted images of the target images I_0 , I_1 , and I_2 , respectively.

It is worth mentioning that the information is embedded into prediction error triplets (PETs) in Section IV. The construction

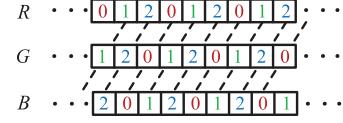


Fig. 4. The construction of PETs in three channels by using the oblique line with the same labels.

of PETs is crucial for the embedding performance. For the pixel $x_{i,j}$, the prediction error $e_{i,j}$ is calculated as follows:

$$e_{i,j} = x_{i,j} - \hat{x}_{i,j}, \quad (3)$$

where $\hat{x}_{i,j}$ is the predicted value. In previous method [45], the PETs are constructed by using three prediction errors located at (i, j) in the three channels, expressed as $(e_{i,j}^r, e_{i,j}^g, e_{i,j}^b)$. However, in the proposed image division method, the pixels at the same location (i, j) in three channels are marked with different labels, as shown in Fig. 4. To explore the correlation between three channels, we construct PETs with prediction errors along the oblique line with the same labels, represented as $(e_{i,j}^r, e_{i,j-1}^g, e_{i,j-2}^b)$.

B. Design of the DPN

In the prediction part, the design of CNN-based predictor is important for improving prediction accuracy. In previous CNN-based methods [19], [20], [21], [22], more neighboring pixels within the current channel are adopted for prediction. Different from these works, the proposed DPN not only utilizes more neighboring pixels in the current channel but exploits the correlation between three channels. Fig. 5 provides the overview of the proposed DPN. The proposed DPN consists of three main components: the key feature extraction part, the significant feature extraction part, and the image prediction part, as shown in Fig. 5 (a).

Different from the basic blocks in [20] and [21], Fig. 5 (b) shows the structure of the feature block (FeatureBlock) used in the key feature extraction part, in which $\lfloor K/2 \rfloor$ convolution operations are connected in series to construct receptive fields of $K \times K$ by only using 3×3 sized kernel [52]. The leaky rectified linear unit (LeakyReLU) with default parameters is used as the activation function. For pixel-wise prediction, the paddings P are all set to 1×1 and the strides S are all set to 1×1 . The channel size is set to $C = 32$ to reduce the computational complexity. In this part, the key feature is constructed by combining different high-dimensional feature maps from FeatureBlocks with receptive fields of 3×3 , 5×5 , and 7×7 , respectively.

To exploit the high correlation between three channels, the significant feature extraction part is proposed for compensation. Three convolution operations and two LeakyReLU activations are connected in series, in which the kernel sizes of three convolution operations are set to 5×5 , 7×7 , and 3×3 , respectively. Correspondingly, the paddings P are set to 2×2 , 3×3 , and 1×1 to facilitate pixel-wise prediction. Furthermore, the strides S are all set to 1×1 , and the channel size is set to $C = 32$. Following this, the attention mechanism [53], referred to as the “SELayer”, is introduced to focus on the significant regions of the extracted features. With the skip

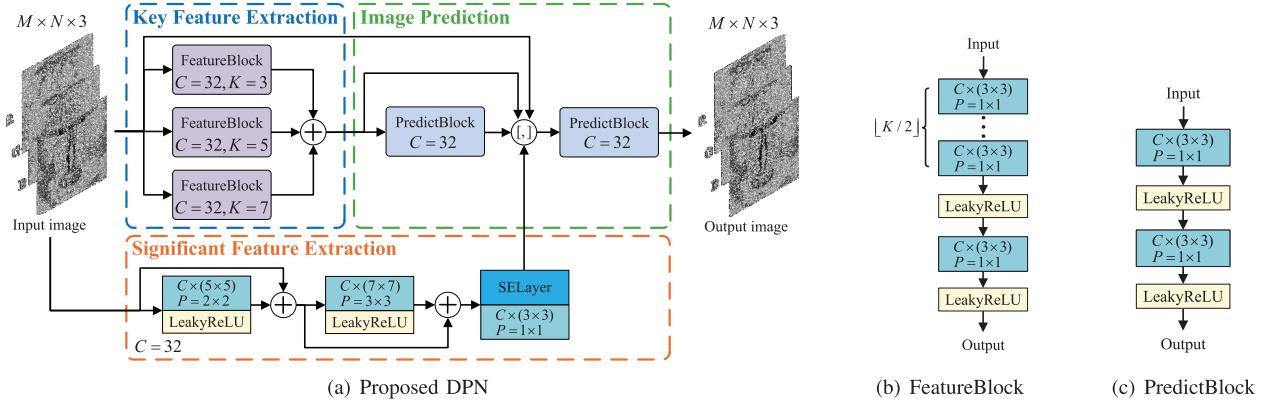


Fig. 5. Overview of the proposed DPN: (a) the structure of the proposed DPN, (b) the structure of FeatureBlock, and (c) the structure of PredictBlock.

connections, the extracted features are fully utilized. In this way, the extracted features are titled as the significant features, which are used to improve the prediction performance.

In the image prediction part, the key features and significant features are used to generate the predicted image. The image prediction part is composed of two concatenated prediction blocks. Fig. 5 (c) shows the structure of the prediction block (PredictBlock), where two convolution operations are connected in series with LeakyReLU as the activation function. Specifically, the kernel sizes, paddings, and strides are set to 3×3 , 1×1 , and 1×1 , respectively, with a channel size of $C = 32$. By employing PredictBlocks, the extracted high-dimensional feature maps are concatenated channel-wise using skip connections to generate the predicted images.

In the proposed DPN, the high-dimensional feature maps are first extracted from the key feature extraction part. After that, the correlation between the three channels is extracted for compensation. Finally, the image prediction part integrates the key features and the significant features to generate the predicted images. Compared with classical predictors, the proposed DPN can use more neighboring pixels in the current channels. Additionally, unlike traditional CNN-based predictors, the DPN exploits the correlation between the three channels. This comprehensive approach enables the DPN to achieve high prediction performance.

C. Train of the DPN

To train the proposed DPN, the train set and the validation set are built by randomly selecting 3,000 images and 750 images from the COCO dataset [54], respectively. The test set comprises the images from the USC-SIPI dataset [55] and the Kodak dataset [56], including 30 standard color images. Before training, the color images in the train and validation sets are resized to $M \times N$ by using bicubic interpolation [57], while the images in the test set remain unchanged. Specifically, we set $M \times N$ to 512×512 . The loss \mathcal{L} for the proposed DPN is formulated as follows:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^n (I_{p_i} - I_{t_i})^2 + \lambda \|\omega\|^2, \quad (4)$$

where n is the number of images, I_p is the predicted image, I_t is the target image, λ is the weight decay, and ω represents

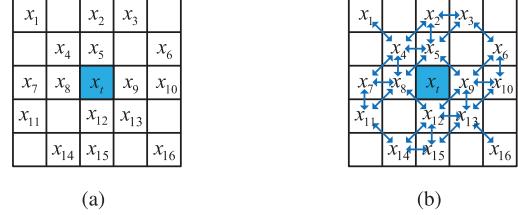


Fig. 6. Illustration of calculating the local complexity in each channel: (a) the context pixels for local complexity calculation and (b) the absolute errors between the context pixels with bi-directional arrows.

all the weights in the proposed DPN. The parameter λ is set to 10^{-3} . The proposed DPN is trained by an NVIDIA TITAN V GPU, optimized through the back-propagation [58] and the Adam optimizer [59].

IV. EMBEDDING PART

To embed information efficiently, the local complexities are first calculated to select the PETs with small absolute values. Next, the 3D PEH is decomposed to demonstrate the property of symmetry, making it feasible to design the 3D mapping strategy. Finally, the implementation details are provided.

A. Local Complexity

In the literature, the local complexity is used to select prediction errors with small absolute values for improving the embedding performance [60]. Several methods for calculating local complexity have been proposed, including the local variance [8], forward variance [61], error energy estimation [62], and local absolute error [26], etc.

In this paper, the local complexity LC of each PET is calculated as the sum of the local complexities of the three prediction errors:

$$LC = L_r + L_g + L_b \quad (5)$$

where L_r , L_g , and L_b are the local complexities of the prediction errors in the red, green, and blue channels, respectively.

In this paper, the local complexity of each prediction error is calculated by using neighboring pixels within the $H \times H$ sized block in the current channel. Based on the proposed

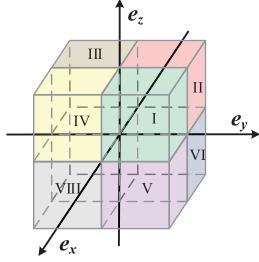


Fig. 7. The distribution of 3D PEH, in which eight quadrants are marked with Roman numerals.

image division method, $\lfloor \frac{2}{3}H^2 \rfloor$ neighboring pixels can be used for this calculation, where the symbol $\lfloor \cdot \rfloor$ denotes the floor function. For example, when $H = 5$, Fig. 6 (a) shows the distribution of neighboring pixels labeled as x_i , where $1 \leq i \leq \lfloor \frac{2}{3}H^2 \rfloor$. The local complexity L_k , for $k \in \{r, g, b\}$, is calculated as follows:

$$L_k = \frac{1}{(H-1)^2} \left(\sum_{i=1}^{\lfloor \frac{2}{3}H^2 \rfloor} \sum_{j=i+1}^{\lfloor \frac{2}{3}H^2 \rfloor} |x_i - x_j| + \sum_{i=1}^{\lfloor \frac{2}{3}H^2 \rfloor} |x_i - \hat{x}_t| \right), \quad (6)$$

where \hat{x}_t is the predicted value of the target pixel x_t . In Eq. 6, the calculation of local complexity comprises two parts. The first part calculates the absolute errors of pixels x_i and x_j , which satisfy $\|\text{loc}(x_i) - \text{loc}(x_j)\|_2^2 \leq 2$. Here, $\text{loc}(\cdot)$ denotes the location of the pixel. For example, the locations of x_5 and x_8 in Fig. 6 (a) are $\text{loc}(x_5) = (2, 3)$ and $\text{loc}(x_8) = (3, 2)$, respectively. To better describe the first part of Eq. 6, each bi-directional arrow in Fig. 6 (b) represents two pixels used to calculate the absolute errors. In the 5×5 sized block, 28 absolute errors can be generated in this first part.

The second part in Eq. 6 calculates the absolute errors between the neighboring pixels x_i and the predicted value \hat{x}_t , revealing the relationship between prediction error and local complexity. In the 5×5 sized block, 16 absolute errors can be generated in this second part. By combining these two parts in Eq. 6, the local complexity L_k , for $k \in \{r, g, b\}$, of each prediction error can be determined. Based on Eq. 5 and Eq. 6, the local complexities of PETs can be calculated.

B. Decomposition of 3D PEH

After counting the PETs, the 3D PEH can be constructed. The symmetry of 3D PEH is then utilized to design the 3D mapping strategy. To analyze this symmetry, we first decompose the 3D PEH into the PET quadrants, PET layers, and PET planes.

- **PET Quadrant.** Since each prediction error in a PET (e_x, e_y, e_z) can be either positive or negative, the 3D PEH consist of 8 quadrants (i.e., $2^3 = 8$), as shown in Fig. 7. For example, the PET $(1, 2, 3)$, $(1, -2, 3)$, and $(1, 2, -3)$ belong to the first, second, and fifth quadrants, respectively. Though the PETs are distributed across these eight quadrants, each PET (e_x, e_y, e_z) can be projected to the first quadrant as $(p(e_x), p(e_y), p(e_z))$, where $p(e_x) \geq 0$,

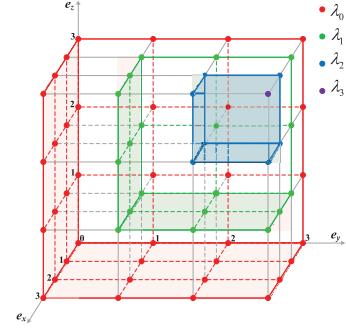


Fig. 8. The distributions of PET layers $\lambda_i, 0 \leq i \leq 3$ in the first quadrant.

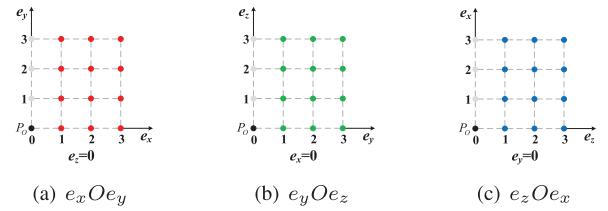


Fig. 9. Three PET planes of the PET layer λ_0 in the first quadrant.

$p(e_y) \geq 0$, and $p(e_z) \geq 0$. The projecting function $p(\cdot)$ is defined as follows:

$$p(e) = \frac{|2e + 1| - 1}{2}. \quad (7)$$

According to Eq. 7, if the prediction error $e \geq 0$, the projected version is $p(e) = e$. If the prediction error $e < 0$, the projected version is $p(e) = -e - 1$. For example, the PETs $(-1, 2, -3)$ and $(-2, -3, 0)$ are projected to $(0, 2, 2)$ and $(1, 2, 0)$, respectively.

- **PET Layer.** Furthermore, each quadrant can be decomposed into PET layers. For instance, in the first quadrant, the i -th PET layer λ_i ($0 \leq i \leq 255$) is defined by PETs (e_x, e_y, e_z) that satisfy the condition:

$$\min(e_x, e_y, e_z) == i. \quad (8)$$

In this way, each quadrant can be decomposed into 256 layers. Fig. 8 shows the distribution of PET planes λ_i for $0 \leq i \leq 3$ in the first quadrant. For example, the PETs $(1, 0, 3)$ and $(2, 2, 0)$ belong to layer λ_0 , the PETs $(2, 1, 3)$ and $(2, 2, 1)$ belong to layer λ_1 , and the PETs $(4, 2, 2)$ and $(2, 3, 4)$ belong to layer λ_2 .

- **PET Plane.** Take the PET layer λ_0 in the first quadrant for example. Each PET layer can be decomposed into 3 PET planes. As shown in Fig. 9, the points marked in red, green, and blue correspond to the PET plane $e_x Oe_y$, $e_y Oe_z$, and $e_z Oe_x$, respectively. Specifically, these three PET planes share the same plane origin P_0 , which is marked in black. According to this decomposition method, the ordinates of PET layers in Fig. 9 (a), (b), and (c) are the abscissas in Fig. 9 (b), (c), and (a), respectively.

After decomposing the 3D PEH, Section IV-C investigated the symmetry of PET quadrants, layers, and planes to simplify the design of 3D mapping into 2D mapping, as detailed in

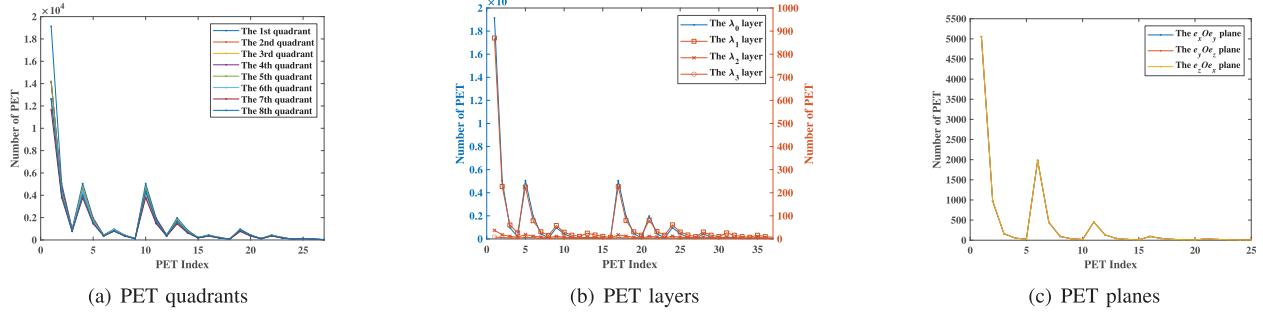


Fig. 10. The frequency changing trends of PETs in the quadrants, layers, and planes.

Section IV-D. To prevent duplicate design in 2D mappings, the points marked in gray in ordinates of Fig. 9 are not considered.

C. Symmetry of 3D PEH

In this section, the symmetry of 3D PEH is demonstrated through statistical experiments. Fig. 10 shows the average frequency-changing trends of PETs across different quadrants, layers, and planes, in which the compared PETs are generated by using the images from the validation and test sets.

Fig. 10 (a) shows the statistical distribution of PETs in eight quadrants. For the first quadrant, the frequency-changing trend of 27 PETs (i.e., $(0, 0, 0) \sim (2, 2, 2)$) is used for comparison. Similarly, the frequency-changing trends for the other seven quadrants correspond to PETs that can be projected to the positions of $(0, 0, 0) \sim (2, 2, 2)$ using Eq. 7. As shown in Fig. 10 (a), the changing trends of PETs in all eight quadrants are similar, indicating that the PET quadrants are symmetrical.

Based on the symmetry of PET quadrants, the first quadrant is taken for further experiment. Fig. 10 (b) shows the distribution of 37 PETs within the first quadrant of layers λ_i , $0 \leq i \leq 3$. As the layer numbers increase, the numbers of PETs decrease sharply. For clarity, the value range of layer λ_0 is shown on the left ordinate, while the value range of layers λ_i , $1 \leq i \leq 3$ is shown on the right ordinate. As shown in Fig. 10 (b), the symmetry can be verified by the similar frequency-changing trends of PETs in four PET layers.

Given the similarly distributed PET layers, the layer λ_0 is utilized to reveal the symmetry of PET planes. For each plane, the 25 PETs nearest to the plane origin are adopted for comparison. As shown in Fig. 10 (c), the distributions of PETs in three planes are close to each other, indicating that the PET planes are symmetrical.

Furthermore, we evaluate the symmetry of 3D PEH by analyzing the correlation coefficients of PETs in quadrants, layers, and planes. Table I presents the correlation coefficients of 27 PETs in 8 quadrants, where all the correlation coefficients exceed 0.990, reflecting high similarity. In Table II, the lowest correlation coefficient is 0.837. As the number of PET layers increases, the number of PETs decreases sharply, leading to a significant drop in the correlation coefficient. Nonetheless, the correlation coefficients of layers λ_i , $0 \leq i \leq 2$ remain high, indicating high symmetry of PET layers. Additionally, the correlation coefficients in Table III exceed 0.998, demonstrating

TABLE I
THE CORRELATION COEFFICIENTS OF EIGHT PET QUADRANTS

	QUA1	QUA2	QUA3	QUA4	QUA5	QUA6	QUA7	QUA8
QUA1	1.000	0.996	0.996	0.996	0.996	0.997	0.991	0.996
QUA2	0.996	1.000	0.998	0.996	0.997	0.999	0.992	0.997
QUA3	0.996	0.998	1.000	0.999	0.997	0.998	0.995	0.999
QUA4	0.996	0.996	0.999	1.000	0.997	0.996	0.992	0.999
QUA5	0.996	0.997	0.997	0.997	1.000	0.999	0.992	0.998
QUA6	0.997	0.999	0.998	0.996	0.999	1.000	0.995	0.998
QUA7	0.991	0.992	0.995	0.992	0.992	0.995	1.000	0.996
QUA8	0.996	0.997	0.999	0.999	0.998	0.998	0.996	1.000

TABLE II
THE CORRELATION COEFFICIENTS OF FOUR PET LAYERS

	Layer λ_0	Layer λ_1	Layer λ_2	Layer λ_3
Layer λ_0	1.000	0.995	0.959	0.869
Layer λ_1	0.995	1.000	0.941	0.837
Layer λ_2	0.959	0.941	1.000	0.969
Layer λ_3	0.869	0.837	0.969	1.000

TABLE III
THE CORRELATION COEFFICIENTS OF THREE PET PLANES

	Plane e_xOe_y	Plane e_yOe_z	Plane e_zOe_x
Plane e_xOe_y	1.000	0.997	0.999
Plane e_yOe_z	0.997	1.000	0.998
Plane e_zOe_x	0.999	0.998	1.000

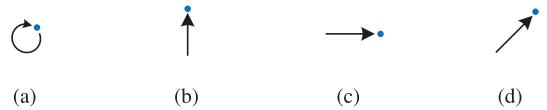


Fig. 11. Four possible inverse mapping ways.

high similarity among PET planes. In this way, the similar distribution of PETs in quadrants, layers, and planes confirms the symmetry of the 3D PEH.

D. Design of 3D Mapping

Based on the symmetry, we propose a new efficient 3D mapping strategy by simplifying the design of 3D mappings into 2D mappings. Take the PET plane e_xOe_y for illustration. Three tricks are adopted to facilitate the design. Firstly, the mapping ways of PETs are designed by using inverse mapping rather than forward mapping. Secondly, the maximum modification of each prediction error is limited to 1, meaning that there are

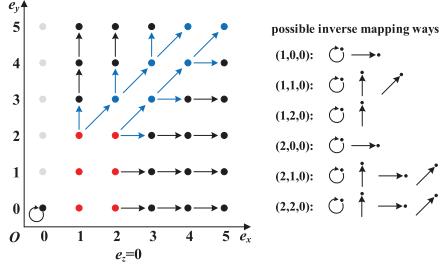


Fig. 12. The basic mapping of the plane e_xOe_y , in which all the possible inverse mapping ways for 6 important PETs are shown on the right side.

only 4 possible inverse mapping ways for each PET, as shown in Fig. 11. Finally, the adaptive method is applied to find the mapping ways of 6 important PETs marked in red, while the mapping ways of other PETs are fixed, as shown in Fig. 12. The possible inverse mapping ways of 6 important PETs can be formulated as:

$$\left\{ \begin{array}{l} f_{inv}^{2D}(1,0,0) = \{(1,0,0), (0,0,0)\} \\ f_{inv}^{2D}(1,1,0) = \{(1,1,0), (1,0,0), (0,0,0)\} \\ f_{inv}^{2D}(1,2,0) = \{(1,2,0), (1,1,0)\} \\ f_{inv}^{2D}(2,0,0) = \{(2,0,0), (1,0,0)\} \\ f_{inv}^{2D}(2,1,0) = \{(2,1,0), (2,0,0), (1,1,0), (1,0,0)\} \\ f_{inv}^{2D}(2,2,0) = \{(2,2,0), (2,1,0), (1,2,0), (1,1,0)\}, \end{array} \right. \quad (9)$$

where $f_{inv}^{2D}(u)$ is the set of possible inverse mapping ways for PET u in the PET plane e_xOe_y . To guarantee reversibility, some inverse mapping ways should be removed, resulting in the generation of 271 reversible 2D mappings.

In the 3D mappings, the theoretical average embedding capacity of each PET u is formulated as $\log_2 |f^{3D}(u)|$, where $|f^{3D}(u)|$ represents the number of forward mapping ways. To embed binary information, $|f^{3D}(u)|$ must be the positive integer exponent power of 2. However, some PETs may not meet the above requirements. Take the PET $(0,0,0)$ for example. If the inverse mapping ways of $(1,0,0)$ and $(1,1,0)$ in plane e_xOe_y are from $(0,0,0)$, after extending 2D mapping into 3D mapping, the inverse mapping ways of PETs in plane e_yOe_z (i.e., $(0,1,0)$ and $(0,1,1)$) and plane e_zOe_x (i.e., $(0,0,1)$ and $(1,0,1)$) will also originate from $(0,0,0)$. In addition, the inverse mapping way of $(0,0,0)$ also comes from itself. Therefore, the PET $(0,0,0)$ has 7 possible forward mapping ways, and the average embedding capacity is $\log_2 7$ bits, making it unsuitable for embedding binary information.

To embed binary information, each PET in plane e_xOe_y should have at most 2 forward mapping ways. Fig. 13 shows the change of forward mapping ways when extending 2D mapping into 3D mapping. In Fig. 13 (a), if PET $(0,0,0)$ has 1 forward mapping way, the extended 3D mapping also has 1 forward mapping way. If PET $(0,0,0)$ has 2 forward mapping ways, the extended 3D mapping has 4 forward mapping ways. For other PETs in the plane e_xOe_y , the forward mapping ways in the 2D mapping remain the same in the 3D mapping, as shown in Fig. 13 (b). In this way, the average embedding capacity of each PET is 0, 1, or 2 bits, making it suitable for embedding binary information.

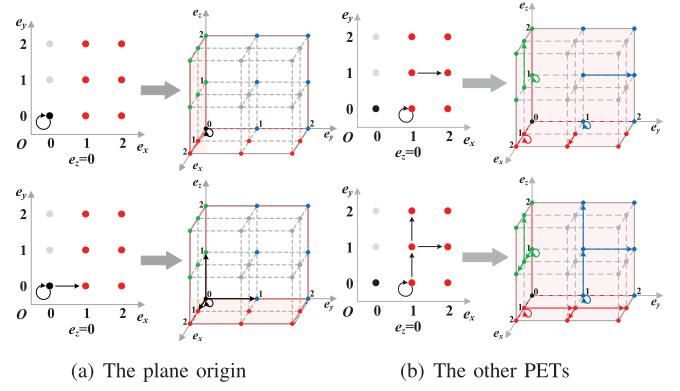


Fig. 13. Illustration of the change in the numbers of forward mapping ways when extending a 2D mapping into a 3D mapping.

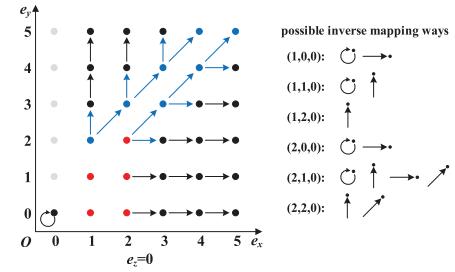


Fig. 14. The refined basic mapping of the plane e_xOe_y , in which the possible inverse mapping ways for 6 PETs are listed on the right side.

After limiting the number of forward mapping ways, Fig. 14 shows the basic 2D mapping. The possible inverse mapping ways are refined as follows:

$$\left\{ \begin{array}{l} f_{inv}^{2D}(1,0,0) = \{(1,0,0), (0,0,0)\} \\ f_{inv}^{2D}(1,1,0) = \{(1,1,0), (1,0,0)\} \\ f_{inv}^{2D}(1,2,0) = \{(1,1,0)\} \\ f_{inv}^{2D}(2,0,0) = \{(2,0,0), (1,0,0)\} \\ f_{inv}^{2D}(2,1,0) = \{(2,1,0), (2,0,0), (1,1,0), (1,0,0)\} \\ f_{inv}^{2D}(2,2,0) = \{(2,1,0)(1,1,0)\}. \end{array} \right. \quad (10)$$

Therefore, 15 reversible 2D mappings can be obtained, constructing the 2D mapping set $\mathcal{F} = \{f_j^{2D}, j = 1, \dots, 15\}$. Among the possible 2D mapping in \mathcal{F} , finding the optimal 2D mapping follows the optimization method:

$$\begin{cases} \min_{f^{2D} \in \mathcal{F}} & ED(f^{2D}) \\ \text{subject to} & EC(f^{2D}) \geq PL, \end{cases} \quad (11)$$

where PL is the payload of information. $ED(f^{2D})$ and $EC(f^{2D})$ are the embedding distortion and embedding capacity of the 2D mapping f^{2D} , respectively. The calculation of $ED(f^{2D})$ and $EC(f^{2D})$ are formulated as follows:

$$ED(f^{2D}) = \sum_{u \in \Phi} H(u) \frac{\sum_{v \in f^{2D}(u)} \|v - u\|^2}{|f^{2D}(u)|}, \quad (12)$$

and

$$EC(f^{2D}) = \sum_{u \in \Phi} H(u) \log_2 |f^{2D}(u)|, \quad (13)$$

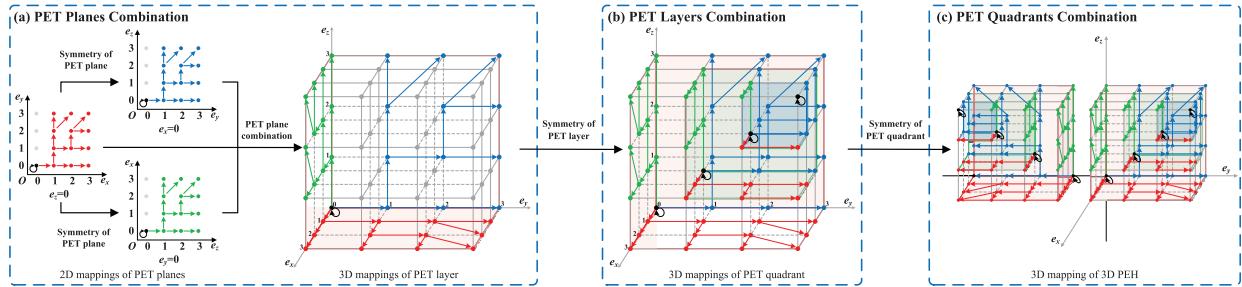


Fig. 15. Illustration of extending a 2D mapping into a 3D mapping by using the symmetry: (a) combine the three 2D mappings of PET planes, (b) combine the 3D mappings of PET layers, and (c) combine the eight 3D mappings of quadrants to build the 3D mapping of 3D PEH.

where Φ is the set of PETs, and $H(u)$ represents the number of the PET u projected to the plane e_xOe_y . During the optimization, the embedding distortion of the 2D mapping f^{2D} is minimized with embedding capacity larger than PL . This approach allows us to achieve the optimal 2D mapping with minimal distortion and sufficient embedding capacity.

After determining the optimal 2D mapping, Fig. 15 shows an example of extending the 2D mapping into the 3D mapping. This involves three main steps. Firstly, the three PET planes share the same 2D mapping based on the symmetry of PET planes, enabling the construction of the 3D mapping for PET layer λ_0 . Secondly, based on the symmetry of PET layers, all the PET layers share the same 3D mappings, thus generating the 3D mapping of the first quadrant. Finally, the symmetry of the quadrants ensures that all eight quadrants share identical 3D mappings, thereby completing the 3D mapping design for the 3D PEH. Based on the observed symmetry, this method efficiently extends the 2D mapping into a 3D mapping, thereby generating the 3D mapping for the 3D PEH.

E. Implementation Details

In the RDH community, auxiliary information is important to ensure reversibility. Since the maximum modification of each prediction error is 1, pixels with values 255 and 0 are changed to 254 and 1, respectively. These changed places are marked as the location map, which is then compressed by the arithmetic coding [63]. In addition to the location map, the auxiliary information also includes:

- the optimal 3D mapping ($3 \times \lceil \log_2(15) \rceil = 12$ bits),
- the final embedded pixels ($3 \times \lceil \log_2(M \times N) \rceil$ bits),

where the symbol $\lceil \cdot \rceil$ represents the ceiling function. Since different information payloads result in different final embedded pixels, it is unnecessary to modify all the pixels to generate the location map, particularly those beyond the final embedded pixel. Take the 512×512 sized color image “Peppers” for example. When the embedding capacity is 20,000 bits, the original auxiliary information is 38,111 bits. However, modifying the pixels beyond the final embedded pixels results in a distortion of 21,081 bits. In this way, the final auxiliary information can be reduced to 17,030 bits. This approach can significantly reduce the size of auxiliary information.

To extract the information, the auxiliary information should be obtained first. To this end, the least significant bits (LSB) substitution [64] is used to embed the auxiliary information.

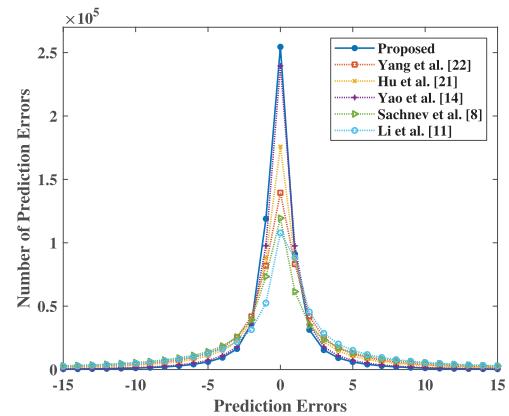


Fig. 16. The histograms of prediction errors of the validation set with the proposed DPN, the prediction methods in Yang and Huang [22], Hu and Xiang [21], Yao et al. [14], Sachnev et al. [8], and Li et al. [11].

Since the LSBs of pixels are replaced with auxiliary information, the replaced bits are combined with information to be embedded by using the proposed RDH method.

V. EXPERIMENTAL RESULTS

In this section, experiments are conducted to evaluate the performance of the proposed RDH method. First, the prediction performance of the proposed DPN is compared with five classical and CNN-based predictors [8], [11], [14], [21], [22]. Next, the embedding performance of the proposed 3D mapping strategy is evaluated with 2D mapping strategy in [27] and 3D mapping strategies in [45], [46], [65]. Finally, the performance of the proposed RDH method is compared with five state-of-the-art RDH methods [14], [34], [45], [46], [47].

A. Performance of the DPN

The prediction performance of the proposed DPN is compared with the CNN-based prediction methods [21], [22] and the classical prediction methods [8], [11], [14]. The prediction errors are adopted for evaluation and quantified by using the histogram, the mean, the variance, and the mean square error (MSE). The following experiments are conducted on the validation set and test set and across different datasets.

Fig. 16 shows the histograms of prediction errors on the validation set. The ordinate represents the number of prediction errors, and the abscissa represents the value of prediction error,

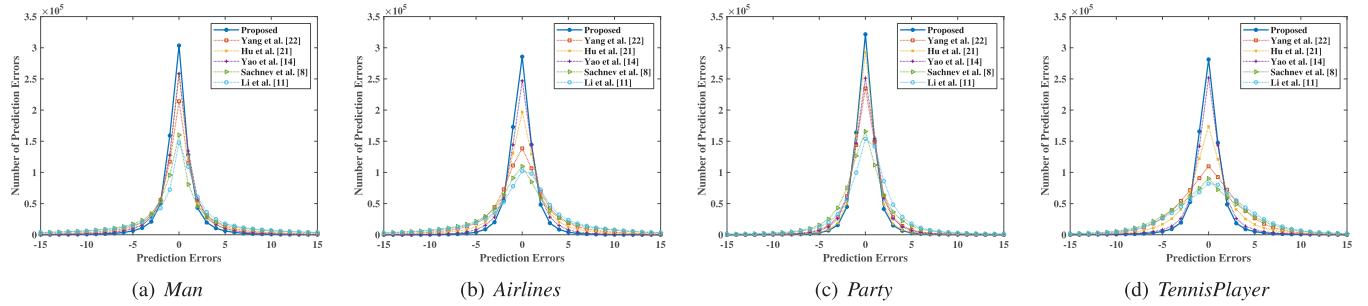


Fig. 17. The histograms of prediction errors of four tested images with the proposed DPN, the prediction methods in Yang and Huang [22], Hu and Xiang [21], Yao et al. [14], Sachnev et al. [8], and Li et al. [11].

TABLE IV

THE MEAN, VARIANCE, AND MSE OF THE ABSOLUTE PREDICTION ERRORS IN THE VALIDATION SET AND THE TEST SET BY USING DIFFERENT PREDICTORS

Predictors	Validation set			Test set		
	Mean	Var	MSE	Mean	Var	MSE
Proposed	1.27	4.65	7.29	1.47	5.11	10.03
Yang et al. [22]	3.61	39.03	59.52	3.66	26.84	44.03
Hu et al. [21]	2.91	28.78	43.09	3.46	23.11	38.73
Yao et al. [14]	1.42	5.81	9.09	1.51	5.77	11.16
Sachnev et al. [8]	5.50	81.85	123.23	5.12	52.41	84.91
Li et al. [11]	5.82	110.34	156.73	5.34	67.50	102.90

TABLE V

THE MEAN, VARIANCE, AND MSE OF THE ABSOLUTE PREDICTION ERRORS IN THE IMAGENET AND PASCAL VOC2012 DATASETS BY USING DIFFERENT PREDICTORS

Predictors	ImageNet			PASCAL VOC2012		
	Mean	Var	MSE	Mean	Var	MSE
Proposed	0.21	0.55	0.91	0.16	0.32	0.51
Yang et al. [22]	0.38	2.09	3.24	0.30	1.27	1.92
Hu et al. [21]	0.38	2.09	3.24	0.30	1.27	1.92
Yao et al. [14]	0.27	0.86	1.45	0.21	0.54	0.87
Sachnev et al. [8]	0.93	10.86	16.92	0.84	9.25	13.82
Li et al. [11]	0.99	14.84	21.67	0.89	12.85	18.03

which is set to $[-15, 15]$ for better visualization. As shown in Fig. 16, the histogram of the proposed DPN is higher and more concentrated compared to other predictors. In addition, Fig. 17 shows the histograms of prediction errors for four standard images from the COCO dataset, where the proposed DPN achieves higher prediction performance than other predictors.

Table IV illustrates the quantitative results of prediction errors. In the validation set, the mean, variance, and MSE of the proposed DPN are 1.27, 4.65, and 7.29, respectively, which are lower than those reported by Yang and Huang [22] (3.61, 39.03 and 59.52), Hu and Xiang [21] (2.91, 28.78 and 43.09), Yao et al. [14] (1.42, 5.81 and 9.09), Sachnev et al. [8] (5.50, 81.85 and 123.23), and Li et al. [11] (5.82, 110.34 and 156.73). Similarly, the results of the proposed DPN in the test set are also lower than those of other predictors.

Furthermore, the prediction performance is compared across datasets. We randomly select 1,000 color images from both the ImageNet dataset [66] and the PASCAL VOC2012 dataset [67]. Table V shows the mean, variance, and MSE of prediction errors generated by different predictors on the ImageNet and PASCAL VOC2012 datasets. The proposed DPN

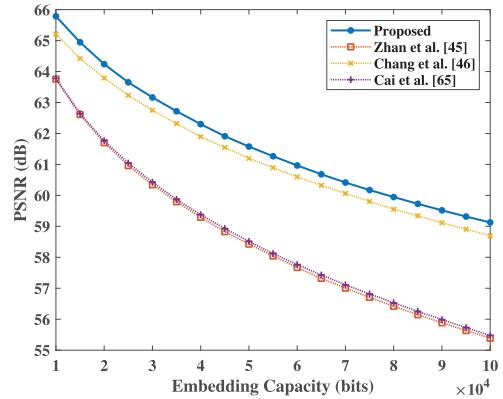


Fig. 18. The average PSNR values (in dB) of the test set in different embedding capacities by using the proposed 3D mapping strategy and the 3D mapping strategies in [45], [46], and [65].

consistently achieves lower mean, variance, and MSE compared to the other predictors.

The reasons for the superior prediction performance of the proposed DPN are as follows. Compared with the classical predictors [8], [11], [14], the proposed DPN can utilize more neighboring pixels in the current channels. In contrast to CNN-based predictors [21], [22], the proposed DPN can exploit the correlation between three channels. Consequently, the prediction accuracy of the proposed DPN is higher than that of both classical and CNN-based predictors.

B. Effectiveness of the Proposed 3D Mapping

The effectiveness of the proposed 3D mapping strategy is analyzed by comparing its embedding performance and the run time with four 2D and 3D mapping strategies [27], [45], [46], [65]. The details are described as follows.

1) Embedding Performance: To evaluate the embedding performance, the proposed 3D mapping strategy and the 3D mapping strategies in [45], [46], and [65] are utilized to construct four RDH methods. The predictor RP serves as the prediction method, and the local complexity calculation method from [8] is employed to select the prediction errors. Consequently, the four RDH methods differ solely in their mapping strategies.

The peak signal-to-noise ratio (PSNR) is adopted to evaluate the embedding performance. Under the same embedding capacity, a higher PSNR indicates better performance.

TABLE VI

THE NUMBER OF MAPPING WAYS GENERATED BY THE PROPOSED 3D MAPPING STRATEGY, THE 2D MAPPING STRATEGY [27] AND THE 3D MAPPING STRATEGY [46] WITH EMBEDDING CAPACITIES OF 20,000 BITS AND 40,000 BITS

Images	20,000 bits			40,000 bits		
	[46]	[27]	Proposed	[46]	[27]	Proposed
Airplane	374	336	15	510	336	15
Baboon	1831	336	15	1678	336	15
House	218	336	15	266	336	15
Lake	1216	336	15	2122	336	15
Oakland	1923	336	15	2209	336	15
Peppers	2690	336	15	2227	336	15
Average	1375	336	15	1502	336	15

TABLE VII

THE RUN TIME (IN SECONDS) OF THE PROPOSED 3D MAPPING STRATEGY, THE 2D MAPPING STRATEGY [27] AND THE 3D MAPPING STRATEGY [46] WITH EMBEDDING CAPACITIES OF 20,000 BITS AND 40,000 BITS

Images	20,000 bits			40,000 bits		
	[46]	[27]	Proposed	[46]	[27]	Proposed
Airplane	2.26	0.45	0.36	4.42	0.39	0.38
Baboon	39.61	0.50	0.26	44.43	1.78	0.28
House	1.08	0.33	0.32	1.67	0.35	0.32
Lake	19.85	0.33	0.30	50.71	0.38	0.30
Oakland	43.44	0.31	0.27	40.86	0.42	0.32
Peppers	53.78	0.39	0.28	45.21	0.57	0.30
Average	26.67	0.38	0.30	31.22	0.65	0.32

Fig. 18 compares the average PSNR of images in the test set, with embedding capacities ranging from 10,000 bits to 100,000 bits in increments of 5,000 bits. The proposed 3D mapping strategy achieves the highest average PSNR. Compared with the 3D mappings in [45] and [65], the proposed one is adaptively changed to suit different images. In addition, compared with the adaptive method in [46], the proposed method has a smaller solution space. Based on the results, the proposed 3D mapping strategy can achieve better performance than previous 3D mapping strategies.

2) *Run Time*: To reflect the efficiency, the run time of embedding information is measured. The compared methods include the 2D mapping strategy in [27] and the 3D mapping strategy in [46]. The operating environment is an Intel Core i9 CPU, and the experimental platform is Matlab R2021a.

Table VI shows the number of possible mapping ways for different mapping strategies with embedding capacities of 20,000 bits and 40,000 bits. The average numbers of the proposed 3D mapping strategy and the method [27] are 15 and 336, respectively. For the method in [46], the numbers vary significantly depending on the situation. The solution space of the method in [46] is much larger than that of the proposed one, indicating higher computational complexity.

Table VII compares the run time of embedding 20,000 bits and 40,000 bits. For the proposed 3D mapping strategy, the average run times are 0.30 seconds and 0.32 seconds respectively, which are shorter than the compared methods in [27] and [46]. According to the results above, the proposed 3D mapping strategy is efficient for embedding information.

C. Quantitative and Qualitative Experiments

To evaluate the embedding performance of the proposed RDH method, we conduct quantitative and qualitative experiments using five state-of-the-art RDH methods for color images. These methods include those proposed by Ou et al. [34], Yao et al. [14], Zhan et al. [45], Chang et al. [46], and Chang et al. [47].

Table VIII shows the PSNR of embedding 10,000 bits, 20,000 bits, and 30,000 bits in 6 standard color images. The average PSNR values of the proposed RDH method are 66.50 dB, 63.01 dB, and 60.69 dB, respectively, which are higher than those of the compared methods. In addition, Fig. 19 shows the changing trend of PSNR, with embedding capacities ranging from 10,000 bits to the maximum capacity in increments of 5,000 bits. As shown in Fig. 19, the proposed RDH method outperforms the state-of-the-art RDH methods in most cases.

In addition, experiments are conducted on the Kodak dataset. Table IX shows the PSNR values of embedding 30,000 bits, 50,000 bits, and 100,000 bits. After embedding 50,000 bits, the proposed RDH method achieves an average PSNR of 63.63 dB, surpassing the results of Ou et al. [34] (60.36 dB), Yao et al. [14] (63.09 dB), Zhan et al. [45] (59.25 dB), Chang et al. [46] (61.24 dB), and Chang et al. [47] (61.91 dB). Fig. 20 shows the changing trend of the average PSNR, with embedding capacities ranging from 10,000 bits to 100,000 bits in increments of 5,000 bits. As shown in Fig. 20, the proposed RDH method consistently achieves higher average PSNRs compared to state-of-the-art RDH methods.

Furthermore, the processing time of the proposed RDH method and the state-of-the-art RDH methods are conducted using the Kodak dataset. The evaluation was performed on the Windows 10 platform with an Intel i9 CPU, utilizing MATLAB R2021a. Table X presents the processing time (in seconds) for different RDH methods with an embedding capacity of 10,000 bits. Notably, the processing time of Zhan et al. [45] is 1.90 seconds, the lowest among the compared methods based on the fixed 3D mapping strategy. The proposed RDH method has a processing time of 4.10 seconds, which is lower than the methods in Ou et al. [34], Yao et al. [14], Chang et al. [46], and Chang et al. [47]. These results indicate that the proposed RDH method achieves a good balance between embedding performance and processing time.

D. Performance Across Datasets

The performance of the proposed RDH method across datasets is evaluated on the ImageNet and the PASCAL VOC2012 datasets. Fig. 21 and Fig. 22 show the changing trend of the average PSNR on the ImageNet and the PASCAL VOC2012 datasets, respectively. The proposed RDH method consistently outperforms the compared RDH methods in terms of average embedding performance.

In addition, the performance of the proposed RDH method is evaluated on various image types, including the high-resolution DIV2K dataset [68], the natural image noise dataset (NIND) [69], and the describable textures dataset (DTD)

TABLE VIII

PSNR (IN dB) OF 6 CLASSICAL COLOR IMAGES OF THE TEST SET GENERATED BY THE PROPOSED RDH METHOD AND FIVE STATE-OF-THE-ART RDH METHODS [14], [34], [45], [46], [47] WITH EMBEDDING CAPACITY OF 10,000 BITS, 20,000 BITS, AND 30,000 BITS

Images	10,000 bits						20,000 bits						30,000 bits					
	[34]	[14]	[45]	[46]	[47]	Proposed	[34]	[14]	[45]	[46]	[47]	Proposed	[34]	[14]	[45]	[46]	[47]	Proposed
Airplane	68.13	68.52	67.97	69.28	67.83	69.92	64.89	65.33	64.68	66.07	64.98	66.61	62.78	63.28	62.85	63.93	63.13	64.51
Baboon	60.58	65.08	60.23	60.65	61.75	64.73	56.81	60.51	56.69	57.46	58.37	59.75	54.49	56.79	54.47	54.99	55.79	56.39
House	68.90	69.12	69.06	69.69	68.14	70.35	65.97	66.05	66.08	66.70	65.29	67.35	64.09	64.21	64.25	64.89	63.51	65.58
Lake	64.33	66.31	64.83	65.33	65.34	67.42	60.28	62.72	61.03	61.71	61.69	63.42	58.03	60.04	58.36	59.05	59.04	60.43
Oakland	60.78	64.08	59.68	59.45	59.56	64.04	58.71	61.48	56.68	56.94	56.66	61.34	57.13	59.57	54.71	55.13	54.58	59.56
Peppers	57.39	60.70	58.00	57.88	57.61	62.56	55.82	58.24	55.88	56.32	56.25	59.59	54.54	56.57	54.42	55.01	55.15	57.66
Average	63.35	65.63	63.29	63.72	63.37	66.50	60.41	62.39	60.18	60.87	60.54	63.01	58.51	60.08	58.18	58.83	58.53	60.69

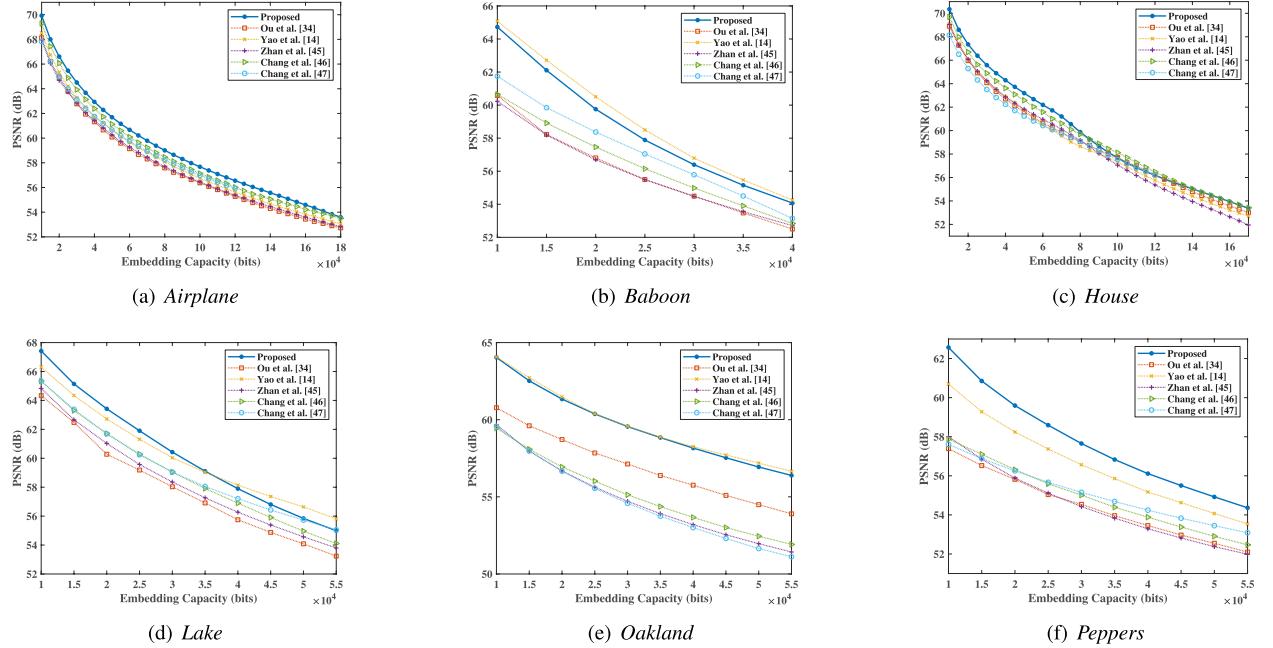


Fig. 19. Performance comparison among the proposed RDH method and the state-of-the-art works [14], [34], [45], [46], [47].

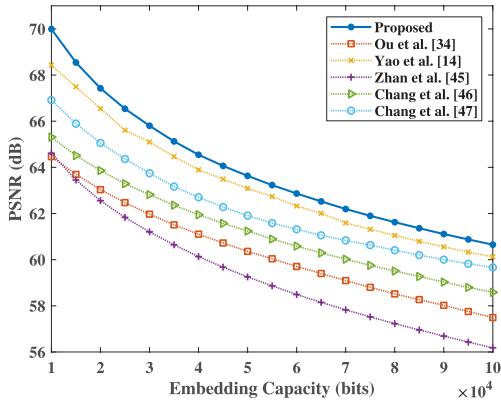


Fig. 20. The average PSNR values (in dB) of the Kodak dataset in different embedding capacities by using the proposed RDH method and the state-of-the-art works [14], [34], [45], [46], [47].

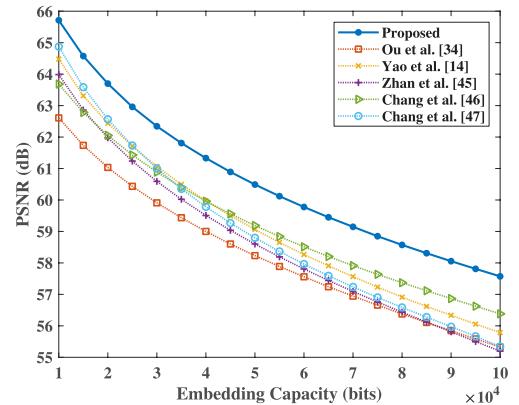


Fig. 21. The average PSNR values (in dB) of the ImageNet dataset in different embedding capacities by using the proposed RDH method and the state-of-the-art works [14], [34], [45], [46], [47].

[70]. Specifically, the noisy images are resized to 512×512 color images, while the other images remain unchanged. Fig. 23 shows the embedding performance of the proposed RDH method across these datasets, with embedding capacities ranging from 10,000 bits to 100,000 bits, in increments of 5,000 bits. Based on Fig. 23, we can find that the proposed

RDH method has acceptable embedding performance across different image types. At an embedding capacity of 100,000 bits, the average PSNR of the proposed RDH method on the DIV2K dataset, NIND dataset, and the DTD dataset are 72.79 dB, 60.71 dB, and 58.67 dB, respectively. These experimental results suggest that the proposed RDH method has

TABLE IX

PSNR (IN dB) VALUES OF THE KODAK DATASET (24 IMAGES) GENERATED BY THE PROPOSED RDH METHOD AND FIVE STATE-OF-THE-ART WORKS [14], [34], [45], [46], [47] WITH EMBEDDING CAPACITY OF 30,000 BITS, 50,000 BITS, AND 100,000 BITS

Images	30,000 bits						50,000 bits						100,000 bits					
	[34]	[14]	[45]	[46]	[47]	Proposed	[34]	[14]	[45]	[46]	[47]	Proposed	[34]	[14]	[45]	[46]	[47]	Proposed
kodim01	64.11	66.21	62.98	64.59	63.56	66.73	61.81	63.95	60.12	62.25	61.74	64.31	57.75	60.83	54.73	59.15	60.80	61.06
kodim02	64.41	66.13	63.39	64.87	63.96	66.25	62.25	63.86	61.04	62.73	62.51	64.07	58.44	60.75	57.72	59.51	60.41	60.99
kodim03	63.69	65.45	63.47	64.40	64.63	67.13	62.21	63.60	61.85	62.94	63.05	64.97	59.67	60.88	58.95	60.37	60.71	61.87
kodim04	64.94	66.35	63.18	65.34	64.45	66.74	62.57	64.07	60.83	63.19	62.22	64.45	59.23	60.92	57.54	59.94	59.10	61.31
kodim05	61.99	65.36	62.01	62.73	63.62	65.49	60.39	63.41	59.78	61.07	61.08	63.44	57.36	60.26	56.35	58.47	58.60	60.34
kodim06	58.21	65.17	59.39	60.04	64.91	64.99	57.67	63.05	58.38	59.41	62.92	62.63	56.02	59.17	56.32	57.40	60.48	59.43
kodim07	65.43	66.20	64.24	65.77	64.84	67.16	63.22	64.14	62.02	63.69	62.87	64.85	60.03	61.05	58.74	60.68	59.65	61.70
kodim08	58.65	64.11	58.34	59.93	60.19	64.30	57.45	62.35	56.39	58.93	59.05	61.33	54.63	59.55	52.79	56.05	57.36	59.23
kodim09	64.62	66.36	61.72	64.90	63.77	66.26	62.27	64.22	59.36	62.84	60.79	64.02	58.72	61.14	56.13	59.77	59.08	61.00
kodim10	63.00	66.04	61.16	63.75	62.97	66.18	61.24	64.01	59.08	61.99	61.00	63.93	58.21	60.99	56.05	59.37	59.16	60.97
kodim11	65.74	66.30	64.89	66.41	66.14	67.75	63.59	64.04	62.59	64.37	63.23	65.47	59.98	61.05	58.99	60.76	59.64	62.25
kodim12	61.39	63.77	61.41	62.47	63.94	66.63	60.26	62.49	59.81	61.24	62.69	64.43	58.17	60.14	57.21	59.04	61.12	61.34
kodim13	60.02	64.29	58.27	60.83	59.40	64.75	58.36	62.51	55.27	58.70	58.91	62.92	54.25	59.59	50.16	54.58	55.75	60.05
kodim14	63.74	66.05	61.82	64.08	63.90	66.32	61.66	63.97	59.27	62.14	60.78	64.06	57.97	60.97	55.37	59.33	58.58	60.93
kodim15	58.04	64.36	59.45	59.62	63.87	65.03	57.51	62.71	58.53	58.98	63.16	62.97	56.29	57.59	56.76	57.48	61.91	59.97
kodim16	65.09	66.40	64.14	65.52	66.61	67.07	62.83	64.24	61.81	63.47	63.85	64.74	59.71	61.13	58.57	60.20	59.14	61.54
kodim17	64.52	66.39	62.74	65.16	63.71	66.70	62.37	64.19	60.37	63.06	62.21	64.46	59.23	61.09	57.11	59.97	60.13	61.35
kodim18	61.88	65.41	59.40	62.86	61.12	65.62	59.89	63.32	57.30	61.23	59.24	63.55	56.44	60.19	54.10	58.63	56.84	60.55
kodim19	63.34	66.09	61.48	64.00	62.59	65.61	61.45	63.95	59.35	62.21	60.82	63.61	57.93	60.73	56.26	59.29	59.79	60.78
kodim20	50.77	55.21	52.63	52.54	65.11	60.14	50.67	54.61	52.55	52.40	64.99	59.18	50.42	53.87	52.28	52.19	64.65	57.43
kodim21	62.07	65.99	60.60	62.93	62.49	65.89	60.57	63.87	58.68	61.49	60.63	63.79	57.52	60.78	55.82	58.91	56.61	60.71
kodim22	61.76	65.84	60.56	62.71	63.51	66.04	60.22	63.69	58.63	61.22	60.42	63.85	57.21	60.42	55.75	58.49	58.62	60.38
kodim23	61.93	65.83	61.99	62.65	64.64	66.19	60.65	63.74	60.30	61.40	62.96	64.13	58.36	60.75	57.69	59.04	60.33	61.14
kodim24	57.97	63.05	59.65	59.50	66.07	64.35	57.51	60.13	58.79	58.80	64.82	62.03	56.25	59.12	56.89	57.42	63.54	59.31
Average	61.97	65.10	61.20	62.82	63.75	65.80	60.36	63.09	59.25	61.24	61.91	63.63	57.49	60.12	56.18	58.59	59.66	60.65

TABLE X

THE AVERAGE PROCESSING TIME (IN SECONDS) OF EMBEDDING 10,000 BITS INTO THE KODAK DATASET BY USING THE PROPOSED RDH METHOD AND THE STATE-OF-THE-ART WORKS [14], [34], [45], [46], [47]

	[34]	[14]	[45]	[46]	[47]	Proposed
Time (s)	29.08	26.55	1.90	16.43	89.95	4.10

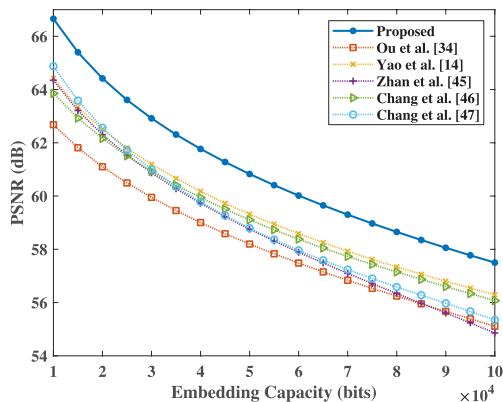


Fig. 22. The average PSNR values (in dB) of the PASCAL VOC2012 dataset in different embedding capacities by using the proposed RDH method and the state-of-the-art works [14], [34], [45], [46], [47].

good generality, making it applicable to a wide range of image types.

In conclusion, the proposed method has achieved satisfactory performance. This achievement is primarily attributed to the high prediction accuracy of the proposed predictor, which effectively reduces the numerical size of prediction errors. Additionally, the efficient 3D mapping strategy in the embedding part simplifies the design of mapping way and reduces embedding distortion.

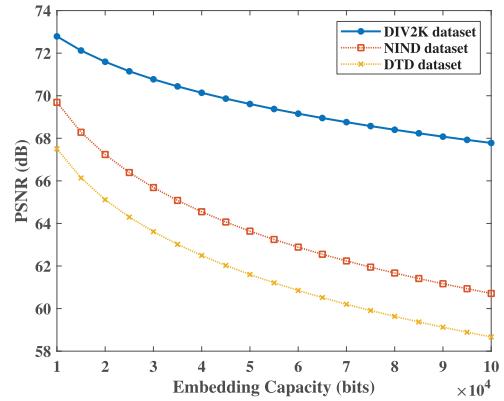


Fig. 23. The average PSNR values (in dB) of the DIV2K, NIND, and DTD datasets in different embedding capacities by using the proposed RDH method.

VI. CONCLUSION

In this paper, we propose a new RDH method for color images by innovating both the prediction part and the embedding part. In the prediction part, each color image is divided into three sets to provide two-thirds of the image information. The proposed DPN extracts key features from the current channel and significant features from other channels to generate predicted images that are highly similar to the target images. Experimental results demonstrate that the proposed DPN achieves high prediction accuracy compared to several classical and CNN-based predictors.

In the embedding part, the PET is constructed by using prediction errors in the oblique lines. The 3D PEH is decomposed into the PET quadrants, PET layers, and PET planes to illustrate the symmetry. This approach simplifies the design of 3D mappings into the design of 2D mappings. Therefore, the

optimal 2D mapping can be extended into the 3D mapping by using symmetry. Compared with the previous 2D and 3D mapping strategies, the proposed 3D mapping strategy is efficient and has satisfactory embedding performance.

Combining the proposed DPN and 3D mapping strategy, the proposed RDH method achieves satisfactory results in embedding information. For six standard color images, the average PSNR is 66.50 dB with an embedding capacity of 10,000 bits. For the Kodak dataset, the average PSNR is 63.63 dB after embedding 50,000 bits, surpassing the compared state-of-the-art RDH methods. Furthermore, results from different kinds of image datasets demonstrate that the proposed RDH method generalizes well. In conclusion, the proposed RDH method is a new step in combining the CNN-based method and efficient mapping strategy, which can improve performance effectively.

REFERENCES

- [1] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Proc. SPIE*, vol. 4314, pp. 197–208, Aug. 2001.
- [2] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
- [3] Y. Shi, X. Li, X. Zhang, H. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.
- [4] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
- [5] S. Mukherjee and B. Jana, "A novel method for high capacity reversible data hiding scheme using difference expansion," *Int. J. Natural Comput. Res.*, vol. 8, no. 4, pp. 13–27, Oct. 2019.
- [6] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
- [7] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 3, pp. 873–882, Sep. 2011.
- [8] V. Sachnev, H. Joong Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, Jul. 2009.
- [9] I. F. Jafar, K. A. Darabkh, R. T. Al-Zubi, and R. A. Al Na'mneh, "Efficient reversible data hiding using multiple predictors," *Comput. J.*, vol. 59, no. 3, pp. 423–438, Mar. 2016.
- [10] W.-J. Yang, K.-L. Chung, and H.-Y.-M. Liao, "Efficient reversible data hiding for color filter array images," *Inf. Sci.*, vol. 190, pp. 208–226, May 2012.
- [11] J. Li, X. Li, and B. Yang, "Reversible data hiding scheme for color image based on prediction-error expansion and cross-channel correlation," *Signal Process.*, vol. 93, no. 9, pp. 2748–2758, Sep. 2013.
- [12] I.-C. Dragoi and D. Coltuc, "Local-prediction-based difference expansion reversible watermarking," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1779–1790, Apr. 2014.
- [13] Y. Yang, T. Zou, G. Huang, and W. Zhang, "A high visual quality color image reversible data hiding scheme based on B-R-G embedding principle and CIEDE2000 assessment metric," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1860–1874, Apr. 2022.
- [14] H. Yao, C. Qin, Z. Tang, and Y. Tian, "Guided filtering based color image reversible data hiding," *J. Vis. Communun. Image Represent.*, vol. 43, pp. 152–163, Feb. 2017.
- [15] Y. Peng et al., "Color image reversible data hiding based on multi-channel synchronized histogram," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 10, Dec. 2023, Art. no. 101804.
- [16] N. Mao, H. He, F. Chen, L. Qu, H. Amirpour, and C. Timmerer, "Reversible data hiding for color images based on pixel value order of overall process channel," *Signal Process.*, vol. 205, Apr. 2023, Art. no. 108865.
- [17] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1138–1150, 2020.
- [18] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "WISERNet: Wider separate-then-reunion network for steganalysis of color images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2735–2748, Oct. 2019.
- [19] T. Luo, G. Jiang, M. Yu, C. Zhong, H. Xu, and Z. Pan, "Convolutional neural networks-based stereo image reversible data hiding method," *J. Vis. Communun. Image Represent.*, vol. 61, pp. 61–73, May 2019.
- [20] R. Hu and S. Xiang, "CNN prediction based reversible data hiding," *IEEE Signal Process. Lett.*, vol. 28, pp. 464–468, 2021.
- [21] R. Hu and S. Xiang, "Reversible data hiding by using CNN prediction and adaptive embedding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 10196–10208, Dec. 2022.
- [22] X. Yang and F. Huang, "New CNN-based predictor for reversible data hiding," *IEEE Signal Process. Lett.*, vol. 29, pp. 2627–2631, 2022.
- [23] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.
- [24] W. Zhang, X. Hu, X. Li, and N. Yu, "Recursive histogram modification: Establishing equivalency between reversible data hiding and lossless data compression," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, Jul. 2013.
- [25] Y. Hu, H.-K. Lee, K. Chen, and J. Li, "Difference expansion based reversible data hiding using two embedding directions," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1500–1512, Dec. 2008.
- [26] B. Ou, X. Li, Y. Zhao, R. Ni, and Y. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.
- [27] B. Ou, X. Li, W. Zhang, and Y. Zhao, "Improving pairwise PEE via hybrid-dimensional histogram generation and adaptive mapping selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2176–2190, Jul. 2019.
- [28] B. Ma and Y. Q. Shi, "A reversible data hiding scheme based on code division multiplexing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 1914–1927, Sep. 2016.
- [29] S. Kim, X. Qu, V. Sachnev, and H. J. Kim, "Skewed histogram shifting for reversible data hiding using a pair of extreme predictions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3236–3246, Nov. 2019.
- [30] J. Yuan, H. Zheng, and J. Ni, "Efficient reversible data hiding using two-dimensional pixel clustering," *Electronics*, vol. 12, no. 7, p. 1645, Mar. 2023.
- [31] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction-error expansion," *Signal Process.*, vol. 93, no. 1, pp. 198–205, Jan. 2013.
- [32] T. Zhang, X. Li, W. Qi, and Z. Guo, "Location-based PVO and adaptive pairwise modification for efficient reversible data hiding," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2306–2319, 2020.
- [33] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.
- [34] B. Ou, X. Li, Y. Zhao, and R. Ni, "Efficient color image reversible data hiding based on channel-dependent payload partition and adaptive embedding," *Signal Process.*, vol. 108, pp. 642–657, Mar. 2015.
- [35] Z. Tang, H. Nie, C.-M. Pun, H. Yao, C. Yu, and X. Zhang, "Color image reversible data hiding with double-layer embedding," *IEEE Access*, vol. 8, pp. 6915–6926, 2020.
- [36] B. Jana, "High payload reversible data hiding scheme using weighted matrix," *Optik*, vol. 127, no. 6, pp. 3347–3358, Mar. 2016.
- [37] J. Biswapat, "Dual image based reversible data hiding scheme using weighted matrix," *Int. J. Electron. Inf. Eng.*, vol. 5, no. 1, pp. 6–19, Sep. 2016.
- [38] J. Biswapat, D. Giri, and M. S. Kumar, "Weighted matrix based reversible data hiding scheme using image interpolation," in *Proc. Int. Conf. Comput. Intell. Data Mining (CIDM)*. Cham, Switzerland: Springer, Dec. 2015, pp. 239–248.
- [39] P. Pal, P. Chowdhuri, and B. Jana, "Weighted matrix based reversible watermarking scheme using color image," *Multimedia Tools Appl.*, vol. 77, no. 18, pp. 23073–23098, Sep. 2018.
- [40] B. Jana, D. Giri, and S. K. Mondal, "Dual image based reversible data hiding scheme using (7,4) Hamming code," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 763–785, Jan. 2018.
- [41] B. Jana, "Reversible data hiding scheme using sub-sampled image exploiting Lagrange's interpolating polynomial," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8805–8821, Apr. 2018.

- [42] P. Pal, B. Jana, and J. Bhaumik, "A secure reversible color image watermarking scheme based on LBP, Lagrange interpolation polynomial and weighted matrix," *Multimedia Tools Appl.*, vol. 80, no. 14, pp. 21651–21678, Jun. 2021.
- [43] P. K. Singh, B. Jana, and K. Datta, "Superpixel based robust reversible data hiding scheme exploiting Arnold transform with DCT and CA," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4402–4420, Jul. 2022.
- [44] D. Hou, W. Zhang, K. Chen, S.-J. Lin, and N. Yu, "Reversible data hiding in color image with grayscale invariance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 363–374, Feb. 2019.
- [45] Y. Zhan, Y. Su, X. Wang, and Q. Pei, "Three-dimensional prediction-error histograms based reversible data hiding algorithm for color images," *Multimedia Tools Appl.*, vol. 78, no. 24, pp. 35289–35311, Dec. 2019.
- [46] Q. Chang, X. Li, and Y. Zhao, "Reversible data hiding for color images based on adaptive three-dimensional histogram modification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5725–5735, Sep. 2022.
- [47] J. Chang, G. Zhu, H. Zhang, Y. Zhou, X. Luo, and L. Wu, "Reversible data hiding for color images based on adaptive 3D prediction-error expansion and double deep Q-network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5055–5067, Aug. 2022.
- [48] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2016, vol. 30, no. 1, pp. 2094–2100.
- [49] F. Song and J. Dongarra, "Scaling up matrix computations on shared-memory manycore systems with 1000 CPU cores," in *Proc. 28th ACM Int. Conf. Supercomputing*, Jun. 2014, pp. 333–342.
- [50] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [51] Ü. V. Çatalyürek, C. Aykanat, and B. Uçar, "On two-dimensional sparse matrix partitioning: Models, methods, and a recipe," *SIAM J. Sci. Comput.*, vol. 32, no. 2, pp. 656–683, Jan. 2010.
- [52] A. Araujo, W. Norris, and J. Sim, "Computing receptive fields of convolutional neural networks," *Distill*, vol. 4, no. 11, p. 21, Nov. 2019.
- [53] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 7132–7141.
- [54] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
- [55] A. G. Weber. (2006). *The USC-SIPI Image Database: Version 5*. [Online]. Available: <https://sipi.usc.edu/database/database.php>
- [56] R. Franzen. (1999). *Kodak Lossless True Color Image Suite*. [Online]. Available: <https://r0k.us/graphics/kodak/>
- [57] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, no. 6, pp. 1153–1160, Dec. 1981.
- [58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [60] S. Xiang and G. Ruan, "Efficient PVO-based reversible data hiding by selecting blocks with full-enclosing context," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2868–2880, May 2022.
- [61] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec. 2011.
- [62] W. Hong, "Adaptive reversible data hiding method based on error energy control and histogram shifting," *Opt. Commun.*, vol. 285, no. 2, pp. 101–108, Jan. 2012.
- [63] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
- [64] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, 2004.
- [65] S. Cai, X. Li, J. Liu, and Z. Guo, "A new reversible data hiding scheme exploiting high-dimensional prediction-error histogram," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2732–2736.
- [66] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [67] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
- [68] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: Dataset and study," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 126–135.
- [69] B. Brummer and C. De Vleeschouwer, "Natural image noise dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1777–1784.
- [70] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3606–3613.



Runwen Hu received the B.S. and Ph.D. degrees from the College of Information Science and Technology, Jinan University, Guangzhou, China, in 2019 and 2024, respectively. He is currently a Post-Doctoral Fellow with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. His current research interests include reversible data hiding and robust data hiding.

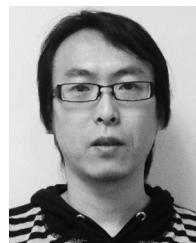


Yuhong Wu received the B.S. and M.S. degrees from the College of Information Science and Technology, Jinan University, Guangzhou, China, in 2020 and 2023, respectively. His current research interests include reversible data hiding.



Shijun Xiang received the B.S. degree from Chang'an University, China, in 1997, the M.S. degree from Guizhou University, China, in 2000, and the Ph.D. degree from Sun Yat-sen University, China, in 2006. From 2006 to 2007, he was a Post-Doctoral Researcher with Korea University, Seoul, South Korea. He is currently a Full Professor with the College of Information Science and Technology, Jinan University, Guangzhou, China. He has authored or co-authored over 100 peer-reviewed articles, including IEEE TRANSACTIONS ON PATTERN

ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and IEEE TRANSACTIONS ON MULTIMEDIA. His current research interests include robust watermarking, reversible data hiding, and AIGC security.



Xiaolong Li (Member, IEEE) received the B.S. degree from Peking University, China, in 1999, the M.S. degree from École Polytechnique, France, in 2002, and the Ph.D. degree in mathematics from the ENS de Cachan, France, in 2006. From 2007 to 2016, he was a Post-Doctoral Fellow and a Researcher with Peking University. He is currently a Professor with the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests include image processing and information hiding.



Yao Zhao (Fellow, IEEE) received the B.S. degree from the Radio Engineering Department, Fuzhou University, Fuzhou, China, in 1989, the M.E. degree from the Radio Engineering Department, Southeast University, Nanjing, China, in 1992, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding and video analysis and understanding. He was named a Distinguished Young Scholar by the National Science Foundation of China in 2010 and was elected as a Chang Jiang Scholar of Ministry of Education of China in 2013.