

Article

# Efficient Reversible Data Hiding Using Two-Dimensional Pixel Clustering

Junying Yuan <sup>1,2</sup>, Huicheng Zheng <sup>1,\*</sup> and Jiangqun Ni <sup>1</sup>

<sup>1</sup> School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510275, China; issjqni@mail.sysu.edu.cn (J.N.)

<sup>2</sup> Guangzhou Nanfang College, Guangzhou 510970, China

\* Correspondence: zhenghch@mail.sysu.edu.cn

**Abstract:** Pixel clustering is a technique of content-adaptive data embedding in the area of high-performance reversible data hiding (RDH). Using pixel clustering, the pixels in a cover image can be classified into different groups based on a single factor, which is usually the local complexity. Since finer pixel clustering seems to improve the embedding performance, in this manuscript, we propose using two factors for two-dimensional pixel clustering to develop high-performance RDH. Firstly, in addition to the local complexity, a novel factor was designed as the second factor for pixel clustering. Specifically, the proposed factor was defined using the rotation-invariant code derived from pixel relationships in the four-neighborhood. Then, pixels were allocated to the two-dimensional clusters based on the two clustering factors, and cluster-based pixel prediction was realized. As a result, two-dimensional prediction-error histograms (2D-PEHs) were constructed, and performance optimization was based on the selection of expansion bins from the 2D-PEHs. Next, an algorithm for fast expansion-bin selection was introduced to reduce the time complexity. Lastly, data embedding was realized using the technique of prediction-error expansion according to the optimally selected expansion bins. Extensive experiments show that the embedding performance was significantly enhanced, particularly in terms of improved image quality and reduced time complexity, and embedding capacity also moderately improved.

**Keywords:** reversible data hiding; rotation-invariant code; two-dimensional pixel clustering; expansion-bin selection; multiple-histogram-based modification



**Citation:** Yuan, J.; Zheng, H.; Ni, J. Efficient Reversible Data Hiding Using Two-Dimensional Pixel Clustering. *Electronics* **2023**, *12*, 1645. <https://doi.org/10.3390/electronics12071645>

Academic Editor: Miin-shen Yang

Received: 24 February 2023

Revised: 22 March 2023

Accepted: 29 March 2023

Published: 30 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the area of data hiding, reversible data hiding (RDH) is capable of completely recovering both the embedded information and the cover image, it specifically effective in data-sensitive applications, where no permanent distortion to the cover media is allowed [1]. In RDH, the embedding performance usually refers to image quality, embedding capacity (EC), and time complexity. Image quality is often measured by the level of image distortion due to data embedding operations. Embedding capacity refers to the maximum amount of information which can be embedded into a cover image. Time complexity measures the computational cost for data embedding procedures.

Many RDH implementations focus on improving the image quality with a considerable embedding capacity. Among them, the most employed techniques include prediction-error expansion (PEE) [2], pixel sorting [3], pixel clustering [4], and multiple-histogram-based modification (MHM) [4]. PEE is usually implemented together with histogram modification to construct prediction-error histograms (PEHs), from which proper expansion bins are selected for histogram shifting and histogram expansion [3–10]. Existing high-performance prediction algorithms include rhombus prediction [3], directionally enclosed prediction (DEP) [5], CNN-based prediction [11–13], etc.

The technique of pixel sorting [3] prefers embedding data into low-complexity (LC) image regions, where prediction errors are assumed to be relatively smaller. As a result, pixel

sorting is usually employed together with PEE for performance improvement [3,8–11]. The most employed sorting criteria is the local complexity. Pixel clustering adheres to the same assumption followed by pixel sorting, but aims to realize cluster-based data embedding by classifying pixels into different groups according to some measurements [4,5,14–16]. Based on pixel clustering, MHM [4,14–23] extends PEE by constructing multiple PEHs based on the pixel clusters and employs PEE for data embedding according to the optimally selected expansion bins.

MHM is well-known for its high image quality, and thus many MHM extensions [10,14–23] have been advanced to further improve its performance. Qin et al. [15] combined the pairwise embedding technique into the framework of MHM and designed a two-dimensional MHM-based method. Hou et al. [16] presented a similar method, but employed a deep neural network (DNN) to generate multiple histograms. Wang et al. [18] employed the fuzzy c-means for pixel clustering instead of directly constructing multiple histograms based on the local complexity. Ou et al. [20] integrated MHM into pixel-value-ordering-based RDH methods. To further exploit the local redundancy, Weng et al. [22] exploited k-means clustering for the construction of multiple histograms, and designed an improved crisscross optimization algorithm to accelerate the procedure of expansion-bin selection. Chang et al. [23] extended MHM into color images with an effective reversible mapping selection mechanism from a three-dimensional PEH.

Note that the techniques of pixel sorting and pixel clustering, as well as the framework of MHM, have been proven to be successful in improving the embedding performance. The common ground they share is that data embedding with finer pixel clustering can better exploit the pixel distribution characteristics. In the light of performance improvement based on pixel clustering, finer pixel clustering by adding more independent clustering factors seems to be a valuable research direction for further performance enhancement, in terms of both image quality and embedding capacity.

In addition to image quality and embedding capacity, time cost is also an important measurement of impact in real-world RDH applications. For optimal data embedding, especially for MHM-based RDH methods, the time cost is usually very heavy due to the process of performance optimization, which tries to select the optimal expansion bins from multiple PEHs. To reduce the runtime of expansion-bin selection, Yuan et al. [24] proposed a technique of fast parameter optimization (FPO), which employs the concept of per-bit distortion (PBC) to simplify the process of expansion-bin selection. As a result, the solution space is significantly reduced, and the runtime of the performance optimization is reduced to tens of milliseconds with only a tiny loss in image quality. Ma et al. [25] proved that optimal expansion bins should have similar per-bit distortions, and developed the expansion-bin determination (EBD) for performance optimization. Their method also reduced the runtime of the performance optimization to about a hundred milliseconds with a tiny loss in image quality. It is worth noting that the reduction in time cost for the expansion-bin selection in these two methods [24,25] is obtained at the expense of a slight reduction in image quality, and the performance is evaluated using the original MHM framework, where only one single clustering factor is employed.

This paper aims to improve the embedding performance of RDH in terms of image quality, embedding capacity, and time complexity. Specifically, the image quality should be improved while increasing the embedding capacity and reducing the time complexity. Firstly, we propose using the rotation-invariant code (RIC) of the local binary pattern [26] as a new factor for pixel clustering. Then, a scheme of two-dimensional pixel clustering is designed with both RIC and LC as the clustering factors, and two of the existing prediction algorithms [3,5,11] are selected to realize cluster adaptive prediction. As a result, two-dimensional PEHs (2D-PEHs) are constructed for two-dimensional MHM (2D-MHM)-based data embedding. Next, a novel technique for expansion-bin selection is proposed to reduce the time complexity of the performance optimization in 2D-PEHs. Lastly, data embedding is realized using PEE for each of the histograms in the 2D-PEHs according to the optimally selected expansion bins.

The rest of this manuscript is organized as follows: The related works are briefly introduced in Section 2. The proposed method is detailed in Section 3, and experimental analysis is provided in Section 4. Section 5 concludes this proposed work.

## 2. Related Works

Some related works are briefly introduced in this section, including those that cover existing high-performance prediction algorithms [3,5,11–13], the framework of MHM [4], and existing fast expansion-bin selection techniques [24,25].

### 2.1. High-Performance Prediction Algorithms

In this section, recently developed high-performance prediction algorithms are introduced, including the rhombus predictor [3], the directionally enclosed predictor [5], and the CNN predictor [11]. The rhombus predictor takes the average of the pixels in the four-neighborhood as the estimated value, and thus can be seen as a fully enclosed predictor. Since pixel estimation is made within the four-neighborhood, rhombus prediction is usually employed together with double-layer-based image segmentation [3,4].

The DEP [5] was used to prove that fully enclosed prediction is not accurate enough in specific image regions, and thus the authors proposed selecting the more accurate estimation from the two directional prediction results. As a result, the DEP selects part of the pixels for data embedding, and thus can be seen as a kind of pixel clustering. The DEP is well-known for its well-improved image quality.

The CNN predictor [11] considers the local features within a certain surrounding image region for pixel estimation, and constructs a CNN module as the image predictor. To keep reversibility, CNN-based prediction is realized as estimating one half of the image according to the pixels in the other half. The CNN predictor has been proven to be more accurate in pixel estimation than the rhombus predictor.

The CNN predictor is later extended to further increase the prediction accuracy [12,13]. Hu et al. [12] extended their first CNN predictor [11] by employing a scheme of four-layer image partition to enhance the prediction accuracy and the technique of adaptive two-dimensional mapping modification to improve the embedding performance. Xie et al. [13] proposed a deeper CNN model, which consists of three modules, namely the feature extraction module, the image prediction module and the reconstruction module. With four-layer image partition, the new CNN model in [13] is proved to be effective in creating a more centrally distracted PEH and in improving the image quality.

### 2.2. Multiple-Histogram-Based Modification

MHM [4] advanced a framework for high-performance data embedding by employing the technique of pixel clustering to divide pixels into multiple groups according to the level of local complexity. In fact, MHM assumes that the local complexity is proportional to the absolute value of the prediction error, and thus prefers embedding information in smooth image regions, where the local complexity tends to be smaller. As a result, MHM is well-known for its high image quality.

In MHM, the secret to its high performance can be explained by the process of performance optimization, which aims to minimize image distortion by optimally selecting multiple expansion bins from each and every PEH. Even though the image quality is significantly improved, MHM is also well-known for its time complexity.

### 2.3. Fast Expansion-Bin Selection

To solve the problem of high time complexity in MHM, two fast expansion bin selection algorithms [24,25] have been proposed and proved to be effective in significantly reducing the runtime of performance optimization to the level of tens of milliseconds in one-dimensional MHM implementations.

The method of fast parameter optimization [24] presented the definition of per-bit distortion and a two-step procedure for expansion-bin selection. Per-bit distortion measures

the average distortion introduced due to PEH shifting and PEH expansion with respect to the selected expansion bins. With this definition, the process of parameter optimization is simplified by selecting expansion bins with lower per-bit distortion. In the two-step expansion-bin selection, the first step tries to detect the upper bound and the lower bound of the solution space, and the second step consists of fine-tuning to select the optimal set of expansion bins.

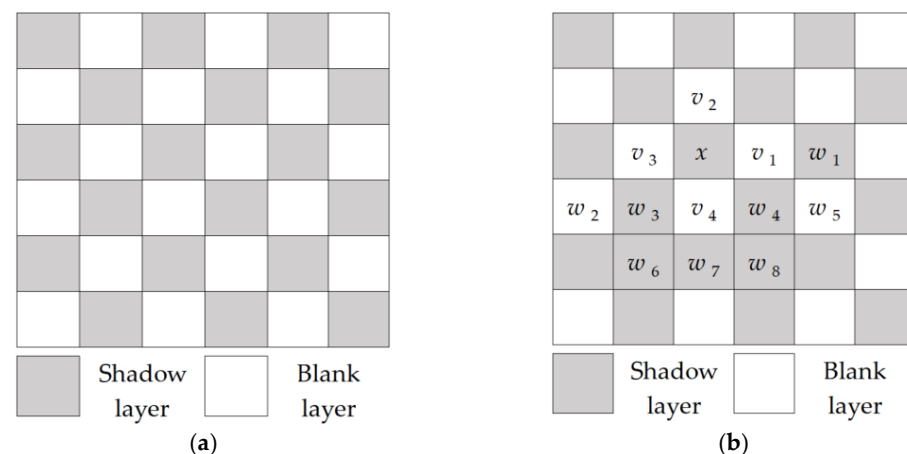
Ma et al. [25] treated the pixel intensity as an amplitude continuous signal, and designed a general form of performance optimization with differentiable objective function and real variables. Then, after theoretical analysis using Lagrange multiplier, a conclusion was reached that optimal expansion bins in the technique of MHM should have similar per-bit distortion.

### 3. The Proposed Method

The superiority of MHM-based RDH methods has been proven to improve image quality. However, in existing MHM implementations, there usually exists only one single clustering factor, and only one simple prediction algorithm is employed to suit all the pixel distributions. Furthermore, existing MHM-based works are well-known to have a high time complexity, which prevents them from being utilized in real-world applications.

In this work, we propose three ideas to improve the embedding performance of MHM in terms of higher image quality, higher embedding capacity, and significantly reduced time complexity. Firstly, the rotation-invariant code of the local binary pattern is designed as a new pixel clustering factor, and the technique of two-dimensional clustering is proposed for the 2D-MHM-based RDH implementation. Then, existing high-performance predictors are exploited for cluster adaptive prediction to improve the image quality and embedding capacity by exploiting their specific advantages. Lastly, a novel algorithm is proposed for the fast expansion-bin selection to significantly reduce the time complexity of the performance optimization for MHM.

Note that the scheme of double-layer image partition is employed in this proposed work to divide the cover image into two non-overlapping layers, a shadow layer and a blank layer, as illustrated in Figure 1b. With the obtained image segmentation result, data embedding is performed in a layer-wise manner.



**Figure 1.** The scheme of double-layer division and the local neighborhood: (a) Double-layer Division; (b) The local neighborhood, where  $x$  is the current pixel,  $v_k$ ,  $1 \leq k \leq 4$  composes the four-neighborhood, both  $v_k$  and  $w_j$ ,  $1 \leq j \leq 4$  forms the local context.

#### 3.1. Rotation-Invariant Code

In this subsection, we exploit the rotation-invariant code [26] of the local binary pattern of pixels in the four-neighborhood to denote the local distribution characteristic. Let  $x$  denote a cover pixel,  $v_k$ ,  $1 \leq k \leq 4$  be the pixels in its four-neighborhood (see Figure 1b), and  $r_i$ ,  $1 \leq k \leq 4$  represent the relationship with  $v_k$ . The local distribution characteristic

for  $x$  can be described using the local binary pattern formed by the relationships between  $r_k$ , as specified by Equation (1),

$$\mathcal{P}(x) = \sum_{k=1}^4 r_k \cdot 2^{k-1} \tag{1}$$

where  $\mathcal{P}(x)$  denotes the local binary pattern, and  $r_i$  is defined by

$$r_k = \begin{cases} 1 & v_k - v_{k+1} \geq 0 \\ 0 & v_k - v_{k+1} < 0 \end{cases} \tag{2}$$

Note that in Equation (2),  $r_4$  is obtained from the relationship between  $r_4$  and  $r_1$ .

With the local binary pattern  $\mathcal{P}(x)$ , the rotation-invariant code (denoted by  $\mathcal{C}$ ) for  $x$  can be obtained [26]. In the four-neighborhood condition, the value of the rotation-invariant code  $\mathcal{C}$  falls in the set of  $S_c = \{1, 3, 5, 7, 15\}$ . Let  $M_c$  denote the number of the rotation-invariant code; thus,  $M_c = 5$ .

### 3.2. Two-Dimensional Pixel Clustering

Traditional pixel clustering is often realized with only a single clustering factor, which is usually the local complexity. With a rotation-invariant code, two-dimensional pixel clustering can be realized by using both the RIC and the LC. Let  $\rho$  denote the local complexity. The LC can be calculated in a traditional way by summing up the local differences in the horizontal and vertical directions, as specified by Equation (3),

$$\rho = |v_1 - w_1| + |w_2 - w_3| + |w_3 - v_4| + |v_4 - w_4| + |w_4 - w_5| + |w_6 - w_7| + |w_7 - w_8| + |v_3 - w_3| + |v_1 - w_4| + |w_1 - w_5| + |w_3 - w_6| + |v_4 - w_7| + |w_4 - w_8|. \tag{3}$$

Let  $n$  ( $1 \leq n \leq M_n$ ) be the level of local complexity, and  $M_n$  be the number of local complexity levels. The value of  $n$  can be obtained by evenly distributing pixels into  $M_n$  pixel clusters. Specifically, pixels are first sorted in an ascending order according to the value of  $\rho$ , and then evenly divided into  $M_n$  clusters. As a result, pixels in each cluster can be treated as the same level, whereas pixels in different clusters can be assigned different complexity levels. From this point of view, the value of  $n$  reflects the level of local complexity for a pixel cluster.

Therefore, each pixel can be labeled with two tags: the rotation-invariant code and the level of local complexity. Consequently, the pixels in each layer (see Figure 1a) can be categorized into a set of two-dimensional clusters according to the specific values of  $\mathcal{C}$  and  $n$ . Let  $G$  denote a pixel cluster; then,  $G(\mathcal{C}, n)$  ( $\mathcal{C} \in S_c$  and  $1 \leq n \leq M_n$ ) can be used to represent a pixel cluster tagged by the same values of  $\mathcal{C}$  and  $n$ . Based on the definition of the rotation-invariant code and the level of local complexity, we can assume that they are independent of each other and are suitable for pixel clustering.

### 3.3. 2D-PEH Construction with Cluster Adaptive Prediction

With the obtained two-dimensional pixel clusters, the prediction errors for the pixels in a cluster  $G(\mathcal{C}, n)$  can be organized into a prediction-error histogram. Let  $h_n^{\mathcal{C}}(e)$  denote the PEH obtained from pixel cluster  $G(\mathcal{C}, n)$ ,  $h_n^{\mathcal{C}}(e)$ , which can be constructed using Equation (4), as specified by

$$h_n^{\mathcal{C}}(e) = \#\{e_i | \mathcal{C}_i = \mathcal{C}, n_i = n, 1 \leq i \leq N\} \tag{4}$$

where  $\#\{\cdot\}$  represents the cardinality;  $N$  is the number of pixels;  $i$  is the index of a pixel; and  $e_i$ ,  $\mathcal{C}_i$ , and  $n_i$  denote the prediction error, the rotation-invariant code, and the level of local complexity for pixel  $x_i$ , respectively. Thus, two-dimensional PEHs are constructed.

Since data embedding is performed in a layer-wise manner and is realized by scanning pixels in a top-to-bottom and left-to-right order, the values of both  $\mathcal{C}$  and  $n$  are reversible at the stage of data extraction and image recovery; thus, the reversibility of the clustering result is guaranteed. Therefore, pixel estimation can be performed in a cluster-wise manner, and the prediction algorithm for each and every cluster can be selected from a batch of

prediction algorithms in order to improve the overall prediction accuracy. Thus, cluster adaptive prediction can be achieved in a 2D-PEH for high-performance data embedding.

### 3.4. Performance Optimization

In PEE-based reversible data hiding, data embedding is realized with two operations, bin shifting and bin expansion. The operation of bin shifting aims to ensure reversibility by creating rooms in the PEH for data embedding, whereas bin expansion utilizes the created room for data embedding via the expansion of the selected bins. Let  $a_n^c, b_n^c$  ( $a_n^c < b_n^c$ ) be a pair of bins selected from the PEH  $h_n^c(e)$ ; then, data embedding in pixel cluster  $G(C, n)$  can be performed using Equation (5), as specified by

$$\acute{e} = \begin{cases} e - 1 & \text{if } e = b_n^c, \\ e - m & \text{if } e < a_n^c, \\ e + m & \text{if } e = a_n^c, \\ e + 1 & \text{if } e > b_n^c, \\ e & \text{otherwise,} \end{cases} \tag{5}$$

where  $\acute{e}$  is the marked version of prediction error  $e$ ,  $e \mp 1$  represents bin shifting, and  $e \mp m$  denotes bin expansion.

Note that in Equation (5), image distortion comes from the two operations of bin shifting and bin expansion. Therefore, when assuming that the secret information follows the independent identical distribution, the overall image distortion, denoted by  $D$ , can be denoted by the number of modified pixels, as specified by Equation (6):

$$D = \sum_{C \in S_c} \sum_{n=1}^{M_n} \left( 0.5 * [h_n^c(a_n^c) + h_n^c(b_n^c)] + \sum_{e < a_n^c} h_n^c(e) + \sum_{e > b_n^c} h_n^c(e) \right) \tag{6}$$

For optimal data embedding, the overall image distortion should be minimized whilst satisfying the requirement of the payload. If  $PS$  denotes the size of the payload, we have to ensure  $PS \geq \sum_{C \in S_c} \sum_{n=1}^{M_n} [h_n^c(a_n^c) + h_n^c(b_n^c)]$ . Then, the problem of performance optimization can be described using Equation (7), as specified by

$$\begin{aligned} & \text{minimize } D \text{ w.r.t.} \\ & PS \geq \sum_{C \in S_c} \sum_{n=1}^{M_n} [h_n^c(a_n^c) + h_n^c(b_n^c)] \end{aligned} \tag{7}$$

Note that in Equations (6) and (7), the introduced image distortion due to the embedding of a given payload is determined by the following parameters:

1. The number of rotation-invariant codes  $M_c$ ;
2. The number of levels of local complexity  $M_n$ ;
3. The selected expansion bins  $a_n^c, b_n^c$ ;
4. The prediction algorithm for each pixel cluster.

In fact, performance optimization specified by Equation (7) is a process of expansion-bin selection from  $M_c \times M_n$  PEHs. If we assume that  $b_n^c$  (or  $a_n^c$ ) is selected from  $M_c$  choices and only two predictors are involved in cluster adaptive prediction, then the solution space for Equation (7) is  $2 \times 2 \times M_e^{M_c \times M_n}$ , the time cost of which is extremely heavy in real-world applications.

### 3.5. Fast Expansion-Bin Selection

To reduce the time cost of expansion-bin selection, the following three rules are provided to simplify the process of performance optimization:

1.  $a_n^c = -b_n^c + b$ , where  $b \in \{1, -1\}$  and is predictor specific;
2. Narrow down the range of  $a_n^c$  and  $b_n^c$  by  $0 \leq b_n^c \leq T$ , where  $T$  is a positive integer;

- Allocate the pixels where  $C = 15$  into the cluster where  $C = 1$ . The reason for this is that too few pixels are tagged with  $C = 15$ , and the two clusters with  $C = 1$  or 15 present a similar PEH distribution.

With these two simplifications, and with the objective of reducing the image quality while improving the embedding capacity and reducing the time complexity, we propose an algorithm for fast expansion-bin selection based on the existing algorithms [24,25]. Before illustrating the algorithm details, the following five definitions are given for clarity.

**Definition 1.** Per bit cost (PBC), which measures the average cost of selecting a pair of expansion bins for data embedding in a pixel cluster  $G(C, n)$ , as specified by Equation (8).

$$PBC(C, n, b_n^C) = \left( \sum_{e > b_n^C} h_n^C(e) + \sum_{e < a_n^C} h_n^C(e) \right) / \left( h_n^C(b_n^C) + h_n^C(a_n^C) \right). \quad (8)$$

**Definition 2.** The PBC matrix is the set of PBCs for data embedding via the expansion of the corresponding expansion bins. Let  $C$  denote a PBC matrix. The shape of a PBC matrix is  $M_C \times M_n \times (T + 1)$ .

**Definition 3.** The matrix of expansion bins is denoted by  $P$ , the shape of which is  $M_C \times M_n$ . Let  $E$  denote the corresponding capacity of  $P$ .

**Definition 4.** The embedding capacity has minimum image distortion (denoted by  $EC_1$ ).  $EC_1$  is obtained by always selecting the cluster-based prediction algorithm with the minimum PBC. The corresponding PBC matrix for  $EC_1$  is denoted by  $C_1$ , and the matrix of corresponding expansion bins is denoted by  $P_1$ . When the size of the payload is no bigger than  $EC_1$ ,  $C_1$  will be employed during expansion-bin selection.

**Definition 5.** The maximum embedding capacity is denoted by  $EC_2$ .  $EC_2$  is obtained by always selecting cluster-based predictor algorithms with the maximum capacity. The corresponding PBC matrix for  $EC_2$  is denoted by  $C_2$ , and the matrix of corresponding expansion bins is denoted by  $P_2$ . When the size of the payload is bigger than  $EC_1$  but no bigger than  $EC_2$ , both  $C_1$  and  $C_2$  will be employed for expansion-bin selection.

With these definitions, the implementation details of the proposed fast expansion-bin selection algorithm are illustrated in Algorithm 1.

---

**Algorithm 1:** Fast bin selection.

---

**Input:**  $PS, EC_1, EC_2, C_1, C_2, P_1, P_2$

**Output:**  $P^*$  and the corresponding prediction algorithms for each selected bin.

**Procedures:**

Step 1. If  $PS \leq EC_1$ , go to Step 2, or else go to Step 5.

Step 2. Sort  $C_1$  in ascending order, and denote the sorted version by  $K_C$ .

Step 3. Select an element  $\kappa_j$  from  $K_C$  using binary searching, find all  $b_n^C$  satisfying  $PBC(b_n^C) \leq \kappa_j$ , and then find the  $P$  with  $E \geq PS$ .

Step 4. Repeat Steps 2 and 3 until the minimum  $\kappa_j$  with  $EC \geq PS$  is found. If  $P^*$  is obtained, go to Step 7.

Step 5. If  $PS \leq EC_2$ , go to Step 6, or else no solution can be detected and process ends.

Step 6. Set  $P^* = P_1$ , replace  $b_n^C$  in  $P^*$  with the corresponding one in  $P_2$  until the corresponding capacity  $E \geq PS$ , and record the corresponding prediction algorithms for each selected bin. Note that  $b_n^C$  is selected from  $P^*$  according to  $PBC(b_n^C)$  in ascending order.

Step 7. Record the corresponding predictor for each bin in  $P^*$ ; process ends.

---

### 3.6. A Simple Implementation

This subsection presents the procedure of data embedding for the proposed method, as illustrated below:

- Step 1. Image pre-processing. Given a cover image, add 1 to the pixels with an intensity of 0 and subtract 1 from the ones with 255. The locations of the modified pixels are recorded in a location map, which is first compressed and then embedding along with the payload [9].
- Step 2. Double-layer image division. The pre-processed image is segmented into two layers, the shadow layer and the blank layer (see Figure 1a). Data embedding is performed in a layer-wise manner.
- Step 3. Perform data embedding in an image layer. Take the shadow layer as an example.
  - 3-1. Image prediction. The shadow layer is individually estimated using two prediction algorithms.
  - 3-2. Obtain the rotation-invariant code (see Section 3.1) and the level of local complexity (see Section 3.2). Then, the two-dimensional pixel clusters are obtained.
  - 3-3. Construct the two-dimensional PEHs (see Section 3.3), and calculate the PBC, the PBC matrix for each and every predictor, the embedding capacity with minimum image distortion ( $EC_1$  and  $P_1$ ), and the maximum embedding capacity ( $EC_2$  and  $P_2$ ) (see Section 3.5).
  - 3-4. Perform Algorithm 1 for fast expansion-bin selection. Then, the optimal parameters are obtained, including the selected bins  $P^*$  and the corresponding prediction algorithm for each pair of selected bins  $a_n^C, b_n^C$ .
  - 3-5. Data embedding is executed using Equation (5). Thus, data embedding is completed for the shadow layer.
- Step 4. After data embedding in both the shadow layer and the blank layer, some necessary side information is recorded in a preserved region, which refers to the boundary pixels in this implementation. The recorded side information includes:
  - The selected bins  $a_n^C, b_n^C$  in  $P^*$ ;
  - The corresponding prediction algorithm;
  - The location of the last modified pixel.

## 4. Experimental Analysis

In this section, the experimental results are presented and discussed to evaluate the embedding performance from three perspectives, namely image quality, embedding capacity, and time complexity. The image quality is measured by the peak-signal-to-noise ratio (PSNR) in dB. The embedding capacity measures the maximum amount of pure payload in bits to be hidden in a cover image. The time complexity refers to the runtime of performance optimization in seconds. Ten typical USC SIPI [27] grayscale images with a size of  $512 \times 512$ , which are commonly used in the area of RDH, are employed as the cover images, as illustrated in Figure 2.

For the performance comparison, we employ three high-performance predictors, rhombus [3], DEP [5], and CNN [11], for cluster adaptive prediction. Then, the combination of two predictors with the least amount of image distortion are employed to further evaluate the image quality, the embedding capacity, and the time complexity. After that, the embedding performance of the proposed work is compared with the typical high-performance RDH implementations, including MHM [4], DEP [4], CNN [5], FPO-APD [24], fast expansion-bins selection (FEBS) [25], and the new CNN-based predictor (CNN New) [13].

In the following subsections, we first illustrate the PEH distributions in Figure 3 and select the preferred combination of prediction algorithms for performance evaluation. Then, we present the image quality in Figures 4 and 5. Lastly, the embedding capacity and the time complexity are provided in Tables 1 and 2, respectively.



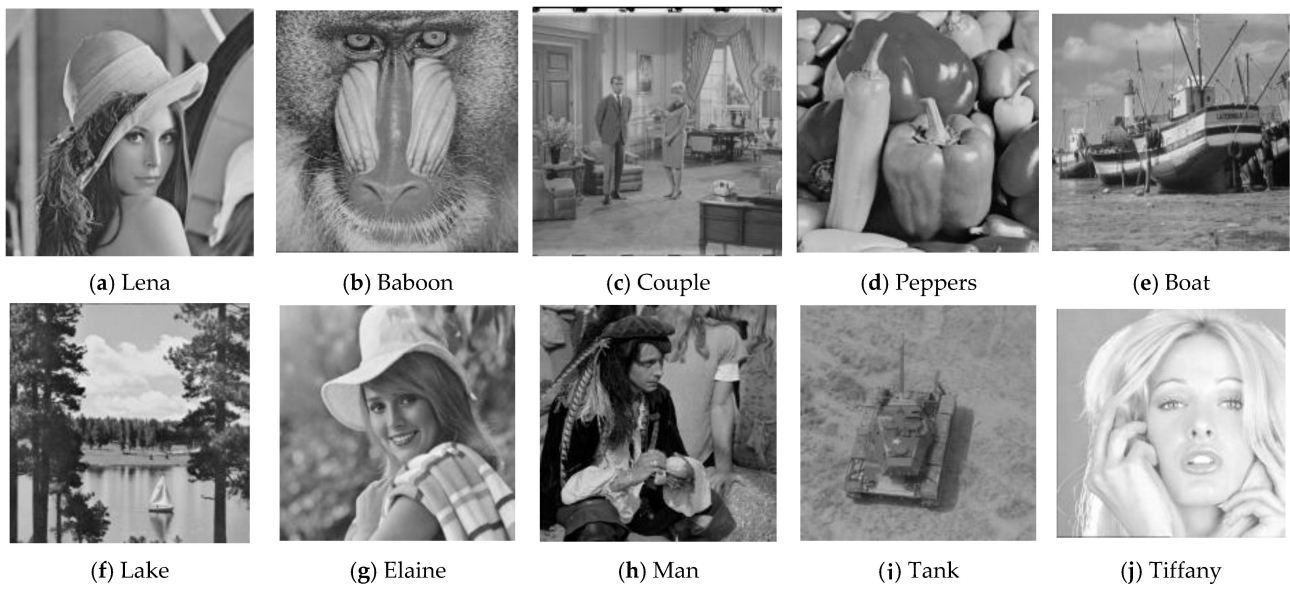


Figure 2. Ten typical test images.

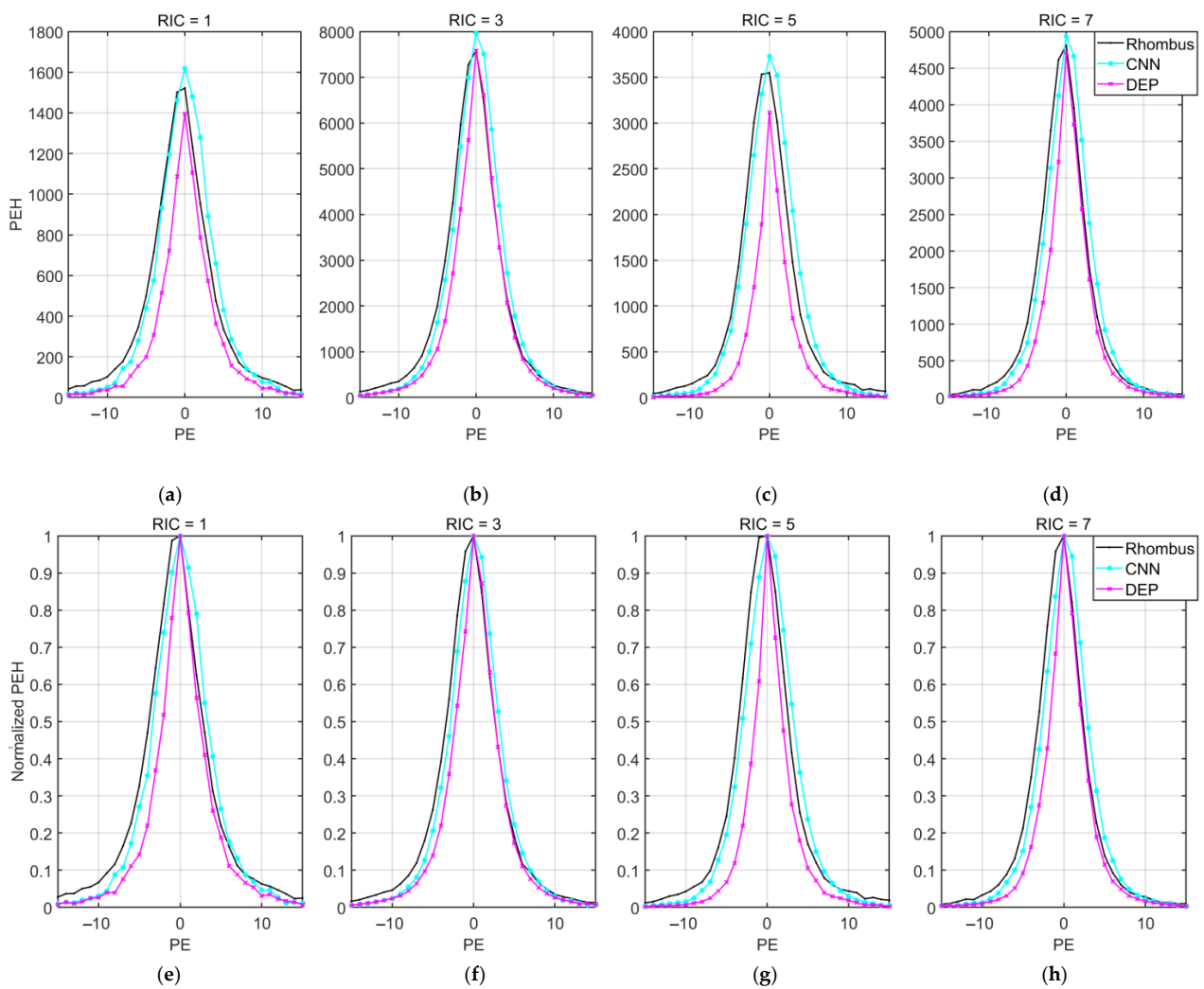


Figure 3. Comparison of PEH distribution obtained using rhombus, DEP, and CNN from the shadow layer of typical image Lena: (a) RIC = 1; (b) RIC = 3; (c) RIC = 5; (d) RIC = 7; (e–h) the normalized version for (a–d), respectively.

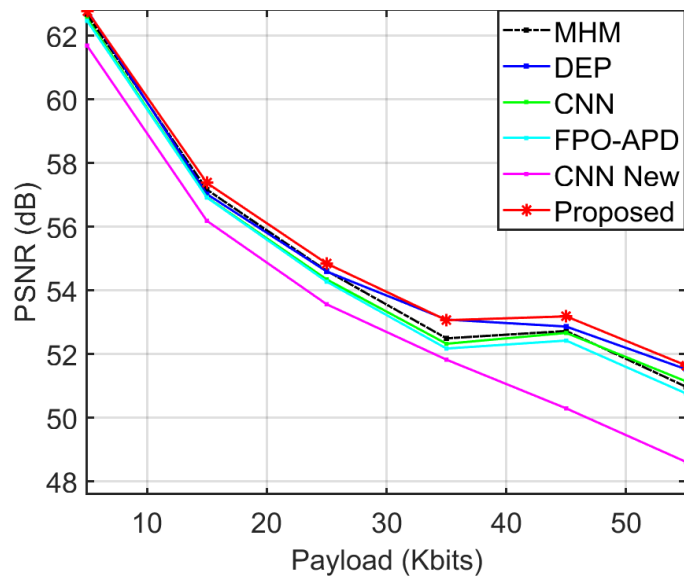
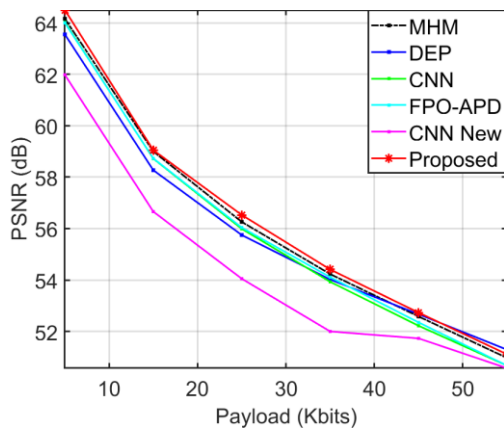
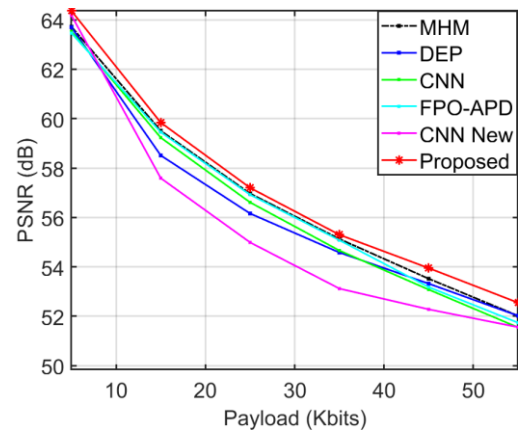


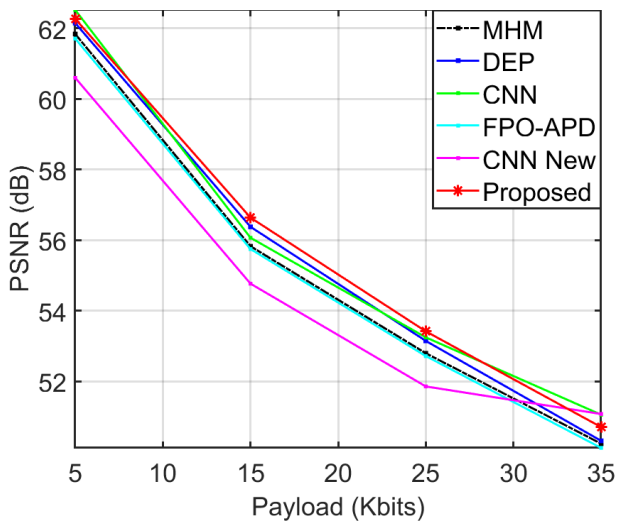
Figure 4. Comparison of average image quality using PSNR.



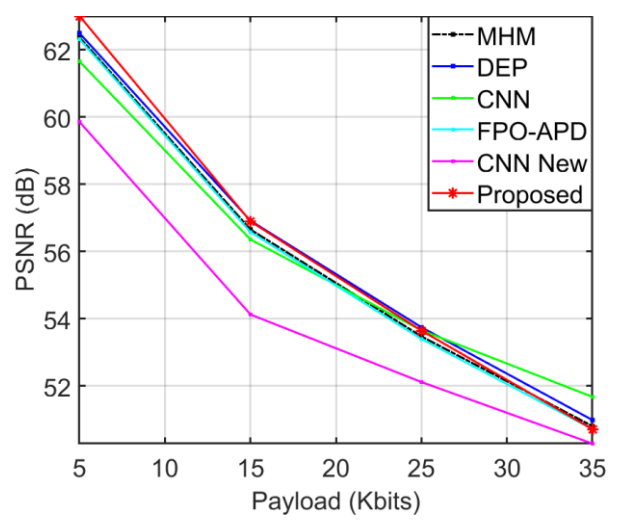
(a)



(b)



(c)



(d)

Figure 5. Comparison of image quality for individual typical test images: (a) Lena, (b) Tiffany, (c) Boat, and (d) Tank.

**Table 1.** Comparison of average PSNRs (dB) with different parameter configurations. The best performance is marked in bold.

Parameters		Size of Payload (Kbits)						
Predictors	$M_n$	5	15	25	35	45	55	65
DEP, CNN	4	57.96	54.74	53.01	51.92	52.85	51.44	50.31
	8	58.50	54.43	52.50	51.94	53.00	51.86	50.39
	16	62.40	56.45	54.12	52.30	53.46	51.99	50.53
DEP, rhombus	4	62.41	57.17	54.72	53.04	53.13	51.57	50.97
	8	62.92	<b>57.46</b>	<b>54.84</b>	<b>53.09</b>	53.15	51.61	<b>51.01</b>
	16	<b>62.93</b>	57.21	54.67	52.99	<b>53.20</b>	<b>51.79</b>	50.95

**Table 2.** Comparison of embedding capacity (bits) of the shadow layer of typical images. The best performance is marked in bold.

Method	Lena	Baboon	Couple	Peppers	Boat	Lake	Elaine	Man	Tank	Tiffany	Average
MHM	34,393	10,997	29,689	21,815	20,508	19,617	18,587	33,612	22,104	37,819	24,914
DEP	30,520	10,614	30,317	20,413	19,045	18,359	17,728	29,816	20,613	35,807	23,323
CNN	<b>35,422</b>	<b>12,348</b>	<b>36,891</b>	21,815	<b>21,262</b>	<b>53,110</b>	<b>18,493</b>	<b>39,337</b>	<b>23,937</b>	<b>39,727</b>	<b>30,234</b>
Proposed	34,567	11,598	32,256	<b>22,045</b>	20,853	19,789	18,737	33,923	22,376	38,745	25,489

#### 4.1. Parameter Configuration

In the proposed two-dimensional pixel clustering, some parameters need configuration in the simple implementation, including the combination of the prediction algorithms and the number of PEHs ( $M_n$ ). The prediction algorithms were selected from the existing high-performance predictors including rhombus, DEP, and CNN. In general, selecting a prediction algorithm with a sharper PEH distribution would produce a high image quality, and another one with higher PEH peak points would generate a larger embedding capacity.

To obtain the combination of a higher image quality and larger embedding capacity, we compared the PEH distributions under different RICs, as illustrated in Figure 3, where Figure 3a–d shows the PEHs obtained from the shadow layer of Lena with RIC = 1, 3, 5, and 7, respectively, and Figure 3e–h shows the normalized version of Figure 3a–d, respectively. It is obvious that the PEHs produced by DEP are more sharply distributed to the origin (see Figure 3e–h). Therefore, DEP is selected as one of the prediction algorithms to preserve the high image quality.

Note that in Figure 3a–d, the two highest peak points ( $e = 0$  and  $e = 1$ , see Figure 3e–h) of both rhombus and CNN are higher than that of DEP, thus, selecting either rhombus or CNN would help in improving the embedding capacity. However, since the embedding capacity is improved by restricting the high image quality, the second prediction algorithm should be selected according to the derived image quality, as illustrated in Table 1.

Note that in Table 1, the predictor combination of DEP and rhombus presents a much better performance than that of the combination of DEP and CNN. With different  $M_n$  configurations, the image quality tends to be better when  $M_n = 8$ . As a result, in the following subsections, we will use DEP and rhombus as the predictor combination, and set  $M_n = 8$  for pixel sorting in the proposed method.

#### 4.2. Image Quality

Figure 4 presents the average PSNR obtained from all ten typical test images, from which it can be observed that the proposed method provides the highest image quality under all payload conditions. For smaller payload conditions when the EC is less than 40 Kbits, the value of the PSNR is slightly improved, but for bigger payload conditions when the EC is larger than 40 Kbits, the improvement of PSNR is more than 0.40 dB. When the size of the payload is 65 Kbits, the PSNR is improved by 1.0 dB. Note that the PSNR for CNN-New [13] is relatively lower, because this method employed the difference-expansion-

based data embedding technique, which presents higher embedding capacity but at the cost of lower image quality.

The comparison of individual typical test images generally echoes the results of the average PSNRs, as illustrated in Figure 5. The only exception is that CNN generates a better PSNR result in the image Tank (see Figure 5d) when the size of the payload is higher; the reason for this is probably that the image contrast is relatively lower and the background is simpler, which leads to a high prediction accuracy when using the CNN predictor.

#### 4.3. Embedding Capacity

The embedding capacity of the shadow layer in typical test images is presented in Table 2, where the largest EC value is marked in bold. It is not difficult to see that CNN provides the best embedding capacity and the EC of the proposed method comes in second, whereas DEP ranks lowest in nearly all images (except in the image Couple). Note that the embedding capacity of CNN-New [13] is provided, because the implementation of CNN-New employs the difference-expansion technique for data embedding, whose embedding capacity of is one bit per pixel.

It is worth noting that the EC of the proposed method is higher compared with that of DEP and rhombus, and the image quality is also improved. This result confirms the effectiveness of the proposed technique of two-dimensional pixel clustering in improving both the image quality and embedding capacity.

Since the focus of FPO-APD [24] is to reduce the time complexity of MHM, the corresponding EC is similar to that of MHM, and thus is not presented.

#### 4.4. Time Complexity

In multiple-histogram-based RDH implementations, performance optimization is usually rather time consuming, which prevent them from being utilized in real-world applications. To reduce the computational power, the proposed method was designed with an expansion-bin selection technique for fast performance optimization in 2D-PEHs.

Table 3 presents the runtime (in seconds) of the parameter optimization for the shadow layer of typical test image, Lena. Note that the number of histograms in both MHM and FPO-APD [24] are set to  $M_n = 16$ , whereas it is set to 32 ( $M_c = 4$ ,  $M_n = 8$ ) in this proposed work. Therefore, the complexity of parameter selection is significantly increased compared to that of MHM, FPO-APD and FEBS [25]. We can see that even though the PEH numbers are doubled, the runtime of the proposed method is still only about 0.10 s, which is good enough for real-world applications.

**Table 3.** Comparison of runtime (in seconds) of parameter optimization for the shadow layer of image Lena. \* The runtime of FEBS is an average value.

Method	Size of Payload (Kbits)				
	5	15	25	35	45
MHM	0.50	0.53	0.54	0.53	0.52
FPO-APD	0.01	0.01	0.02	0.02	0.02
FEBS *	0.05	0.05	0.05	0.05	0.05
Proposed	0.08	0.11	0.09	0.07	0.11

It is worth noting that the performance optimization in both MHM and FPO-APD applies to one-dimensional MHM only, whereas the proposed expansion-bin selection suits both 1D- and 2D-MHM conditions, as well as even higher dimensional cases. This result confirms the effectiveness of the proposed technique of expansion-bin selection. The time complexity for DEP [5] and CNN [11] is similar to that of MHM, and thus, is not illustrated.

#### 4.5. Discussion

The experimental results proved the effectiveness of our proposed work in improving the image fidelity and the embedding capacity, and in reducing the computational complexity. The image fidelity improved because the proposed technique of two-dimensional pixel clustering is effective in creating finer pixel divisions ( $M_c \times M_n = 32$ ) with good intra-cluster correlation, and thus more accurate prediction can be obtained using pattern adaptive prediction result selected via the combination of prediction algorithms. The embedding capacity improved moderately because performance optimization is updated to select expansion bins with higher occurrence frequency when the size of payload exceeds a certain level (see Algorithm 1).

The significantly reduced computational complexity (see Section 4.4) proved the effectiveness of the presented method in performance optimization (see Section 3.4), which firstly designed several concepts based on the concept of per-bit distortion, and then simplified the procedure of expansion-bin selection using dichotomy based on the sorted sequence of per-bit distortion. Note that even though the presented method of performance optimization (see Section 3.4) is implemented to handle optimization in two-dimensional PEH condition, it can also be easily implemented to suit the requirement of one-dimensional or multi-dimensional PEH conditions.

#### 5. Conclusions

This study presented two techniques to improve the embedding performance of multiple-histogram-based modification. The first proposed technique aimed to improve both image quality and embedding capacity by designing a technique of two-dimensional pixel clustering with cluster adaptive prediction. The second technique is the fast expansion-bin selection, which aimed to reduce the time complexity. The performance of the proposed cluster adaptive prediction depends highly on existing prediction algorithms, which is heuristically selected during the combination of prediction algorithms, and thus may not be effective enough in adapting to the characteristics of pixel clusters. In future investigations, efforts will be paid to design more effective cluster adaptive prediction algorithms and the combination of multiple predictors.

**Author Contributions:** Conceptualization, J.Y. and H.Z.; methodology, J.Y. and H.Z.; software, J.Y.; validation, J.Y.; investigation, J.Y.; resources, J.Y. and J.N.; writing—original draft preparation, J.Y.; writing—review and editing, H.Z. and J.N.; supervision, H.Z. and J.N.; project administration, H.Z. and J.N.; funding acquisition, J.Y. and J.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded in part by the Guangdong scientific research promotion project for key construction disciplines under grant number 2021ZDJS132, in part by the Guangdong engineering technology center of regular universities under grant number 2021GCZX001, and in part by the scientific research program of Guangzhou under grant number 202201010098.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Li, X.; Shi, Y.Q.; Zhang, X.; Wu, H.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237.
2. Ni, Z.; Shi, Y.Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
3. Sachnev, V.; Kim, H.J.; Nam, J.; Suresh, S.; Shi, Y.Q. Reversible watermarking algorithm using sorting and prediction. *IEEE Trans. Circuits Syst. Video Technol.* **2009**, *19*, 989–999. [[CrossRef](#)]
4. Li, X.; Zhang, W.; Gui, X.; Yang, B. Efficient reversible data hiding based on multiple histograms modification. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 2016–2027.
5. Chen, H.; Ni, J.; Hong, W.; Chen, T. High-fidelity reversible data hiding using directionally enclosed prediction. *IEEE Signal Process. Lett.* **2017**, *24*, 574–578. [[CrossRef](#)]
6. Ou, B.; Li, X.; Wang, J.; Peng, F. High-fidelity reversible data hiding based on geodesic path and pairwise prediction-error expansion. *Neurocomputing* **2017**, *226*, 23–34. [[CrossRef](#)]

7. Xiao, M.; Li, X.; Wang, Y.; Zhao, Y.; Ni, R. Reversible data hiding based on pairwise embedding and optimal expansion path. *Signal Process.* **2019**, *158*, 210–218. [[CrossRef](#)]
8. He, W.; Cai, Z.; Wang, Y. High-fidelity reversible image watermarking based on effective prediction error-pairs modification. *IEEE Trans. Multimed.* **2021**, *23*, 52–63. [[CrossRef](#)]
9. Hong, W.; Chen, T.S. A local variance-controlled reversible data hiding method using prediction and histogram-shifting. *J. Sys. Softw.* **2010**, *83*, 2653–2663. [[CrossRef](#)]
10. Kaur, G.; Singh, S.; Rani, R.; Kumar, R.; Malik, A. High-quality reversible data hiding scheme using sorting and enhanced pairwise PEE. *IET Image Process.* **2022**, *4*, 1096–1110. [[CrossRef](#)]
11. Hu, R.; Xiang, S. CNN prediction based reversible data hiding. *IEEE Signal Process. Lett.* **2021**, *28*, 464–468. [[CrossRef](#)]
12. Hu, R.; Xiang, S. Reversible data hiding by using CNN prediction and adaptive embedding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 10196–10208. [[CrossRef](#)] [[PubMed](#)]
13. Xie, Y.; Huang, F. New CNN-based predictor for reversible data hiding. *IEEE Signal Process. Lett.* **2022**, *29*, 2627–2631.
14. Wang, W.; Wang, C.; Wang, J.; Bian, S.; Huang, Q. Improving multi-histogram-based reversible watermarking using optimized features and adaptive clustering number. *IEEE Access* **2020**, *8*, 134334–134350. [[CrossRef](#)]
15. Qin, J.; Huang, F. Reversible data hiding based on multiple two dimensional histograms modification. *IEEE Signal Process. Lett.* **2019**, *26*, 843–847. [[CrossRef](#)]
16. Hou, J.; Ou, B.; Tian, H.; Qin, Z. Reversible data hiding based on multiple histograms modification and deep neural networks. *Signal Process. Image Comm.* **2021**, *92*, 116118. [[CrossRef](#)]
17. Wang, J.; Chen, X.; Ni, J.; Mao, N.; Shi, Y.Q. Multiple histograms based reversible data hiding: Framework and realization. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2313–2328. [[CrossRef](#)]
18. Wang, J.; Mao, N.; Chen, X.; Ni, J.; Wang, C.; Shi, Y.Q. Multiple histograms based reversible data hiding by using FCM clustering. *Signal Process.* **2019**, *159*, 193–203. [[CrossRef](#)]
19. Wang, J.; Ni, J.; Zhang, X.; Shi, Y.Y. Rate and distortion optimization for reversible data hiding using multiple histogram shifting. *IEEE Trans. Cybern.* **2016**, *47*, 315–326. [[CrossRef](#)]
20. Ou, B.; Li, X.; Wang, J. Improved PVO-based reversible data hiding: A new implementation based on multiple histograms modification. *J. Vis. Commun. Image Represent.* **2016**, *38*, 328–339. [[CrossRef](#)]
21. Qi, W.; Li, X.; Zhang, T.; Guo, Z. Optimal reversible data hiding scheme based on multiple histograms modification. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 2300–2312. [[CrossRef](#)]
22. Weng, S.; Tan, W.; Ou, B.; Pan, J.-S. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. *Inf. Sci.* **2021**, *549*, 13–33. [[CrossRef](#)]
23. Chang, Q.; Li, X.; Zhao, Y. Reversible data hiding for color images based on adaptive three-dimensional histogram modification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5725–5735. [[CrossRef](#)]
24. Yuan, J.; Zheng, H.; Ni, J. Multiple histograms-based reversible data hiding using fast performance optimization and adaptive pixel distribution. *Comput. J.* **2022**, *bxac109*. [[CrossRef](#)]
25. Ma, S.; Li, X.; Xiao, M.; Ma, B.; Zhao, Y. Fast expansion-bins-determination for multiple histograms modification based reversible data hiding. *IEEE Signal Process. Lett.* **2022**, *29*, 662–666. [[CrossRef](#)]
26. Ojala, T.; Pietikäinen, M. Multiresolution gray-scale and rotation invariant texture classification with local binary pattern. *IEEE Trans. PAMI* **2002**, *24*, 971–987. [[CrossRef](#)]
27. The USC-SIPI Image Database. Available online: <http://sipi.usc.edu/database> (accessed on 1 July 2022).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.