Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 896 – 910

doi:10.3906/elk-1309-62

Prediction-based reversible image watermarking using artificial neural networks

Mahsa AFSHARIZADEH*, Majid MOHAMMADI
Department of Computer Engineering, Faculty of Engineering, Shahid Bahonar University, Kerman, Iran

Received: 09.09.2013 • **Accepted/Published Online:** 20.01.2014 • **Final Version:** 23.03.2016

Abstract: In prediction-based reversible watermarking schemes, watermark bits are embedded in the prediction errors. An accurate prediction results in smaller prediction errors, more efficient embedding, and less distortion for the watermarked image. In this paper, an accurate prediction is made using artificial neural networks. Before the embedding operation, 2 neural networks are trained by the pixel values of the image. Then the trained neural networks predict the pixel values that are used in the embedding operation. Due to the training ability of the neural networks, the prediction will be more accurate than the averaging technique. Experimental results show that the proposed scheme yields superior results compared to several related schemes.

Key words: Reversible watermarking, prediction error expansion, difference expansion, histogram shifting, sorting, artificial neural network

1. Introduction

Nowadays, most information in the world is in digital form. One of the most popular types of digital information are digital images. Digital image watermarking embeds hidden data in the image, known as a watermark, for various applications such as content authentication, copyright protection, and fingerprinting. The embedding operation in watermarking techniques causes certain irreversible modifications in the original image that cannot be removed; therefore, the receiver cannot access the original image before embedding. In sensitive fields, such as military, medical, and astronomical applications, the receiver needs the original image without any changes [1]. Traditional watermarking methods cannot be used for these sensitive applications. Consequently, another kind of watermarking known as reversible watermarking was introduced to solve this problem [1–35]. In reversible watermarking schemes, after the watermark extraction, full recovery of the original image before embedding is possible. So far, several reversible watermarking methods have been presented. Most existing methods belong to three types: difference expansion [1,2,5,8,10,11,16–26], histogram-shifting [3,9,12,14,15,27,28], and prediction-based methods [4,13,29–35]. Difference expansion was introduced by Tian [2]. In the difference expansion method, the watermark bits are embedded by expanding the difference values between the pairs of neighborhood pixels in the image. The histogram-shifting method was introduced by Ni et al. [3]. They used the maximum and minimum points of the histogram for watermark embedding, and the lower bound of the peak signal-to-noise ratio (PSNR) criterion value of the watermarked image versus the original image was higher than 48.13 dB. The prediction-based methods used the prediction for reversible watermarking. One of these methods, a reversible watermarking algorithm using sorting and prediction, was presented by Sachnev et al. [4]. In Sachnev et al.'s work, first the pixel values were predicted by averaging four pixel values in a rhombus

*Correspondence: mafsharizade4@gmail.com

pattern, and then the prediction errors were calculated. These errors were sorted by their local variance values. Finally, watermark bits were embedded in sorted prediction errors with the histogram-shifting technique. When embedding the watermark, if the prediction error value was small, the watermark bit was embedded in the prediction error with the difference expansion technique. Otherwise, the embedding was not performed; only a histogram shift was performed. It was proven that the rhombus pattern prediction scheme is more efficient than the JPEG-LS used by Thodi and Rodriguez [5]. Prediction is a major step in prediction-based reversible watermarking schemes, and an accurate prediction causes less distortion of the watermarked image. Since a powerful prediction algorithm generates smaller prediction errors for embedding, finding a suitable prediction method is an important task in prediction-based reversible watermarking schemes. In this paper, an intelligent powerful prediction scheme is proposed. The paper uses artificial neural networks instead of averaging in a rhombus pattern for prediction. Artificial neural networks simulate the human brain's learning process and hence they have high learning ability. After the training procedure, the neural networks can predict accurately. The rest of this paper is structured as follows: Section 2 is a review of difference expansion, rhombus prediction scheme, sorting technique, and artificial neural networks. Section 3 describes the proposed method, Section 4 presents the experimental results, and Section 5 concludes the paper.

2. An overview of related works

2.1. Difference expansion

One of the most popular methods in reversible watermarking is the difference expansion technique [2]. This method is briefly described below.

Suppose x and y are two pixels of a gray-scale image. The difference and average values of these two pixels can be calculated with Eq. (1).

$$l = \left\lfloor \frac{x+y}{2} \right\rfloor \ ; \ h = x - y \tag{1}$$

The inverse of this transform is computed by Eq. (2).

$$x = l + \left\lfloor \frac{h+1}{2} \right\rfloor \ ; y = l - \left\lfloor \frac{h}{2} \right\rfloor \tag{2}$$

Given that the allowed range for x and y is [0,255] so as to avoid overflow, the difference value h should satisfy Eq. (3).

$$0 \le l + \left\lfloor \frac{h+1}{2} \right\rfloor \le 255 \ ; \ 0 \le l - \left\lfloor \frac{h}{2} \right\rfloor \le 255 \tag{3}$$

Hence, to avoid overflow and underflow, the difference value h must satisfy Eq. (4).

$$|h| \le \min(2 \times (255 - l), 2 \times l + 1) \tag{4}$$

The embedding operation can only be performed for the expandable and changeable difference values. The expandable difference value is defined by Eq. (5).

$$|2 \times h + b| \le \min(2(255 - l), 2 \times l + 1) \tag{5}$$

The changeable difference value is defined by Eq. (6).

$$\left|2 \times \left\lfloor \frac{h}{2} \right\rfloor + b\right| \leq \min(2(255-l), 2 \times l + 1) \tag{6}$$

2.2. Rhombus pattern prediction and sorting techniques

Sachnev et al. proposed a reversible watermarking based on the prediction and sorting technique [4]. In their scheme, first, all pixels in the image are divided into 2 sets: crosses and dots. The pixels belonging to the dot set are used to predict the pixels in the cross set and vice versa. In Sachnev et al.'s work, a cell consists of five pixels, namely a pixel in the center of the cell and four neighborhood pixels in four directions: up, down, left, and right. First, a set of pixels belonging to the cross or dot set are predicted by another set, and then the difference between the original value and the predicted value of the predicted pixels is calculated and referred to as the prediction error. After computing the prediction errors, they are all arranged in ascending order based on their local variance values. Finally, the watermark bits are embedded in the sorted prediction errors with the histogram-shifting technique. In Sachnev et al.'s work, half of the pixels are used to predict and the other half are used for embedding the watermark bits. Hence, the maximum hiding capacity can be 0.5 bits per pixel (bpp). To achieve higher levels of hiding capacity, the double-embedding scheme is used. Double-embedding consists of two procedures: cross-embedding and dot-embedding. First, the cross-embedding procedure is performed. In cross-embedding, the pixels belonging to the cross set are predicted by the pixels belonging to the dot set, and embedding is performed in the prediction errors belonging to the cross set. Dot-embedding starts after cross-embedding. The output of the cross-embedding procedure is the input for the dot-embedding procedure, and the embedded pixels belonging to the cross set are used to predict the pixels in the dot set. The rest of the watermark is embedded in the prediction errors belonging to the dot set. Maximum hiding capacity in the double-embedding scheme can be 1 bpp. Cross and dot sets as well as the cells in cross-embedding and dot-embedding procedures are shown in Figure 1. The cross-embedding procedure is as follows.

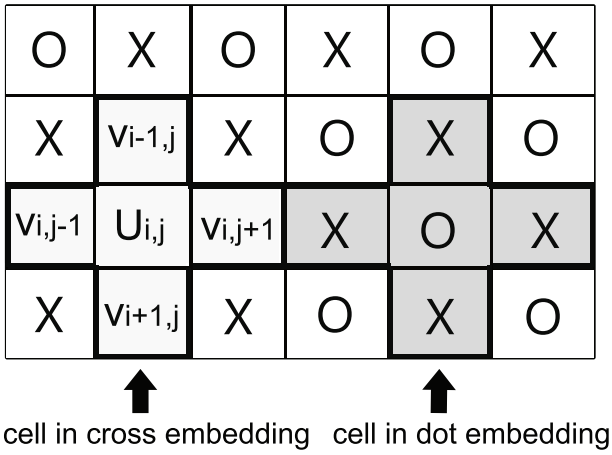

Figure 1. Cells in the cross- and dot-embedding procedures.

In cross-embedding, each pixel belonging to a cross set is predicted by four neighboring pixels belonging to the dot set in the cell, as calculated by Eq. (7).

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor \tag{7}$$

$u_{i,j}$ is the pixel belonging to the cross set in the center of the cell. In Eq. (7), $v_{i,j-1}$, $v_{i+1,j}$, $v_{i,j+1}$, and $v_{i-1,j}$ are the neighboring pixels of $u_{i,j}$ belonging to the dot set. $u'_{i.j}$ is the predicted value of pixel value $u_{i,j}$. The prediction error $d_{i,j}$ is calculated by Eq. (8).

$$d_{i,j} = u_{i,j} - u'_{i,j} \tag{8}$$

The embedding procedure is performed by histogram shifting. In histogram shifting, two negative and positive threshold values T_n and T_p are used. If the prediction error is small, bit $b\epsilon$ {0, 1}is embedded by the difference expansion in the prediction error; otherwise, only histogram shifting is performed and not embedding.

The difference between the expansion and histogram-shifting techniques used in the embedding procedure, is computed by Eq. (9).

$$D_{i,j} = \begin{cases} 2d_{i,j} + b \ if \ d_{i,j} \in [T_n, T_p] \\ d_{i,j} + T_p + 1 \ if \ d_{i,j} > T_p \ and \ T_p \geq 0 \\ d_{i,j} + T_n \ if \ d_{i,j} < T_n \ and \ T_n < 0 \end{cases} \tag{9}$$

Here, $D_{i,j}$ is the prediction error after an expansion or shift. After data-embedding, the original pixel value $\mathrm{u}_{i,j}$ is changed to $\mathrm{U}_{i,j}$ by Eq. (10).

$$U_{i,j} = D_{i,j} + u'_{i,j} \tag{10}$$

The extracting and recovery procedures are as follows.

The original values of the prediction errors $d_{i,j}$ are restored by Eq. (11), and the embedded watermark b is extracted by Eq. (12):

$$d_{i,j} = \begin{cases} \lfloor D_{i,j}/2 \rfloor \ if \ D_{i,j} \in [2T_n, 2T_p + 1] \\ D_{i,j} - T_p - 1 \ if \ D_{i,j} > 2T_p + 1 \ and \ T_p \geq 0 \\ D_{i,j} - T_n \ if \ D_{i,j} < 2T_n \ and \ T_n < 0 \end{cases}, \tag{11}$$

and

$$b = D_{i,j} \bmod 2, D_{i,j} \in [2T_n, 2T_p + 1]. \tag{12}$$

Finally, the original pixel values are recovered by Eq. (13).

$$u_{i,j} = u'_{i,j} + d_{i,j} \tag{13}$$

Sachnev et al. used a sorting technique to reduce the distortion of the watermarked image. Afterwards, the prediction error values ??are calculated for the cells, and then the cells are sorted based on their local variance values in ascending order. The embedding procedure is performed in the sorted cells. In fact, the sorting procedure sorts the cells based on the degree of their smoothness. A small local variance value represents a smooth cell. The sorting technique causes the embedding procedure to start from smoother cells in the image. The local variance value $\mu_{i,j}$ is computed from four neighboring pixels, $\nu_{i,j-1}$, $\nu_{i+1,j}$, $\nu_{i,j+1}$, and $\nu_{i-1,j}$, by Eq. (14):

$$\pi_{i,j} = \frac{1}{4} \sum_{k=1}^{4} (\Delta\nu_k - \Delta\bar{n}u_k)^2 \tag{14}$$

where

$$\Delta\nu_1 = j\nu_{i,j-1} - \nu_{i-1,j}j, \Delta\nu_2 = j\nu_{i-1,j} - \nu_{i,j+1}j,$$

$$\Delta\nu_3 = j\nu_{i,j+1} - \nu_{i+1,j}j, \Delta\nu_4 = j\nu_{i+1,j} - \nu_{i,j-1}j,$$

and

$$\Delta\bar{\nu}_k = (\Delta\nu_1 + \Delta\nu_2 + \Delta\nu_3 + \Delta\nu_4)/4.$$

2.3. Artificial neural network

An artificial neural network simulates the human brain's decision-making and learning processes. It is composed of neurons that perform computing operations and try to simulate the human brain's neural system. All neurons are placed in three types of layers: input, hidden, and output. The backpropagation neural network is a supervised neural network [36]. Figure 2 shows a backpropagation neural network. As shown in Figure 2, each layer has one or more neurons. Each neuron is connected to the adjacent neurons in the neighboring layers. Two neurons in neighboring layers are connected directly to create a connection. Each connection has a weight that determines the degree of the correlation between the two neurons. The purpose of this network is training by input samples and then an accurate prediction on similar inputs. In the backpropagation neural network, the weights are updated by Eqs. (14) and (15):

$$w_{jk}(new) = w_{jk}(old) + \alpha\delta_k y_j, \tag{15}$$

and

$$v_{ij}(new) = v_{ij}(old) + \alpha\delta_j x_i. \tag{16}$$

In Eqs. (14) and (15), w_{jk} is the weight between hidden and output layers, α is the learning rate, δ_k is the error signal between hidden and output layers, and δ_j is the error signal between input and hidden layers. To achieve an optimal artificial neural network structure, the number of layers and the number of neurons in each layer should be determined. Choosing the number of neurons in the hidden layer is an important issue. A very small number of neurons in the hidden layer results in weak learning, and too many neurons in the hidden layer results in a complex neural network. Hence, the neural network stops at the local optimums in the training procedure.

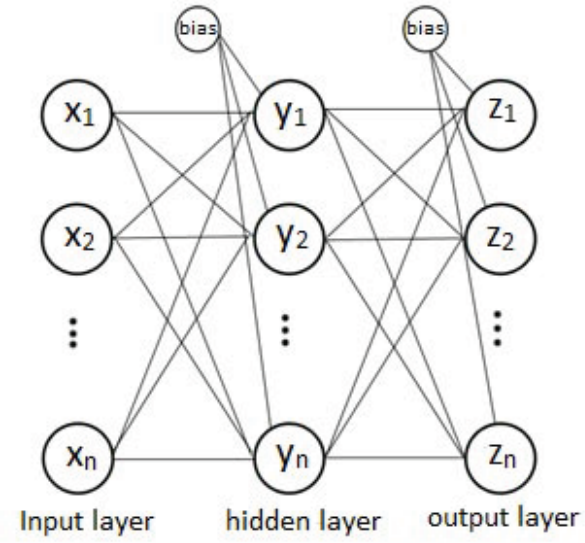

Figure 2. Backpropagation neural network.

3. Proposed method

Choosing an appropriate prediction scheme is one of the most important tasks in reversible watermarking methods. An accurate prediction generates small prediction errors. The watermark bits are embedded into the prediction errors; therefore, small prediction errors improve the embedding performance and reduce the distortion of the watermarked image. Sachnev et al. used the rhombus pattern prediction method [4]. In this method, the value of each pixel in the center of a cell is predicted by averaging four neighboring pixels in the cell. These four pixels are the adjacent pixels in four directions: up, down, left, and right.

In this paper, we also use the rhombus pattern method, although we use two artificial neural networks to predict the pixel values instead of using the averaging technique. Both artificial neural networks used in this paper include 4 neurons in the input layer and one neuron in the output layer. In the cross-embedding procedure, the central cross pixel in each cell is considered as the output for the neural network, and the 4 neighboring dot pixels in each cell are considered as the inputs.

To have an optimal neural network structure, we must determine the number of layers and the number of neurons in each layer. The number of neurons in input and output layers are four and one, respectively. We tested several neural networks with different hidden layers and a different number of neurons in the hidden layers to determine the appropriate neural network structure. We compared the accuracy of all neural networks, and the number of neurons in the hidden layer is set to the number that gives maximum accuracy to the neural network.

We tested the neural networks with one hidden layer where each network has 1–20 neurons, and the network with 18 neurons had the highest accuracy. For two hidden layers, we tested the networks with 1–5 neurons in the hidden layers. The neural network with five neurons in each layer had the highest accuracy, as a higher number of neurons needs more training time. The networks with more than two hidden layers require a long training time.

Each pixel predicted by neural networks is a float number. Since the pixel values are integers, the value predicted by the neural networks should be rounded. The comparison between two neural networks (first neural network with one hidden layer and 18 neurons, and second network with two hidden layers and five neurons in each layer) shows that there is no difference between them after rounding. This means that the predicted values from these two neural network structures are the same, because there is such little difference in the decimal part of the predicted values between these two neural networks that, after rounding, their predicted outputs will be the same. Finally, we chose the network with one hidden layer and 18 neurons in the hidden layer.

In this paper we use the double-embedding technique. The double-embedding technique is composed of cross-embedding and dot-embedding procedures; therefore, we need two neural networks for the embedding procedure. The first is used in cross-embedding to predict the pixel values belonging to the cross set, and the second is used in dot-embedding to predict the pixel values belonging to the dot set. Both are tested with a different number of hidden layers and a different number of neurons in the hidden layers. For both the neural networks, the number of neurons in the hidden layer is set to the number that gives maximum accuracy. We got one hidden layer and 18 neurons in the hidden layer for both the neural networks. For this network, training time is approximately 10 min. The architecture of both the neural networks used in this paper is shown in Figure 3. The architecture of the other appropriate neural network is shown in Figure 4. If we need to train the neural networks for the image, then we can embed different watermarks in the image with the saved trained neural networks. Both neural networks are trained only in the embedding stage and the weight of the neural network is adjusted. These data are used as side information in the extraction stage, and the trained neural networks are used to predict the pixel values.

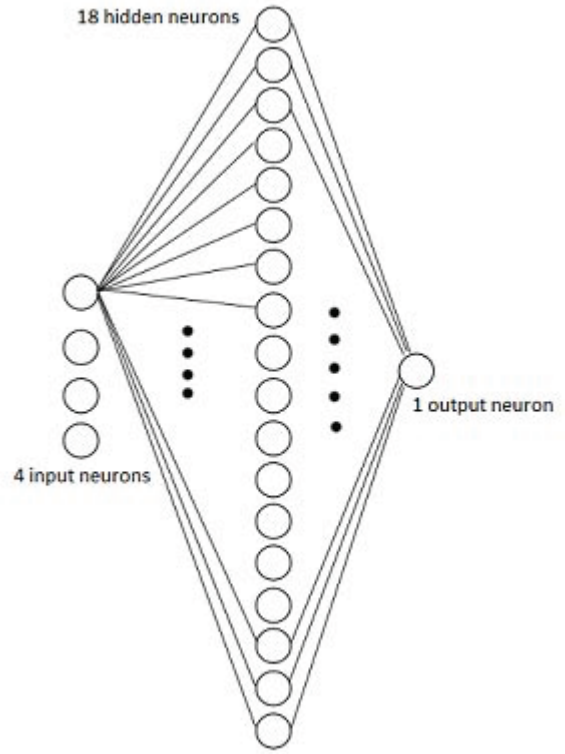

Figure 3. The architecture of both neural networks.

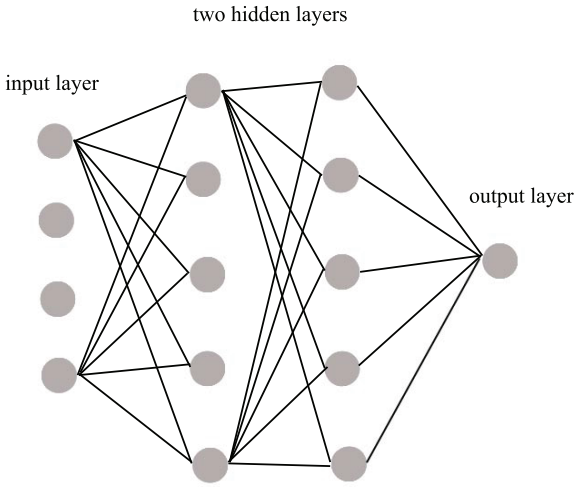

Figure 4. The architecture of another appropriate neural network.

In our proposed method we use the double-embedding technique, which consists of cross-embedding and dot-embedding procedures. The proposed cross-embedding and extracting procedures are described below.

3.1. Proposed embedding procedure

The proposed cross-embedding algorithm is as follows:

Find all the cells whose central pixel belongs to the cross set.

Predict the cross pixel in the center of each cell using the first neural network.

Compute the prediction errors $d_{i,j}$ by Eq. (8).

Compute the local variance values for all cells by Eq. (14).

Sort the cells based on their local variance values.

Skip the first 34 cells of the sorted cells. The 34 least significant bits (LSBs) are collected as a port of the watermark.

Determine threshold values T_n and T_p as presented by Sachnev et al. [4].

Embed the watermark according to the histogram shifting presented by Sachnev et al. [4].

The output of the cross-embedding procedure is the input of the dot-embedding procedure. In the dot-embedding procedure, the embedded pixels belonging to the cross set are used to predict the pixels belonging to the dot set. The dot-embedding procedure is similar to the cross-embedding procedure. The proposed watermark-embedding procedure is shown in Figure 5.

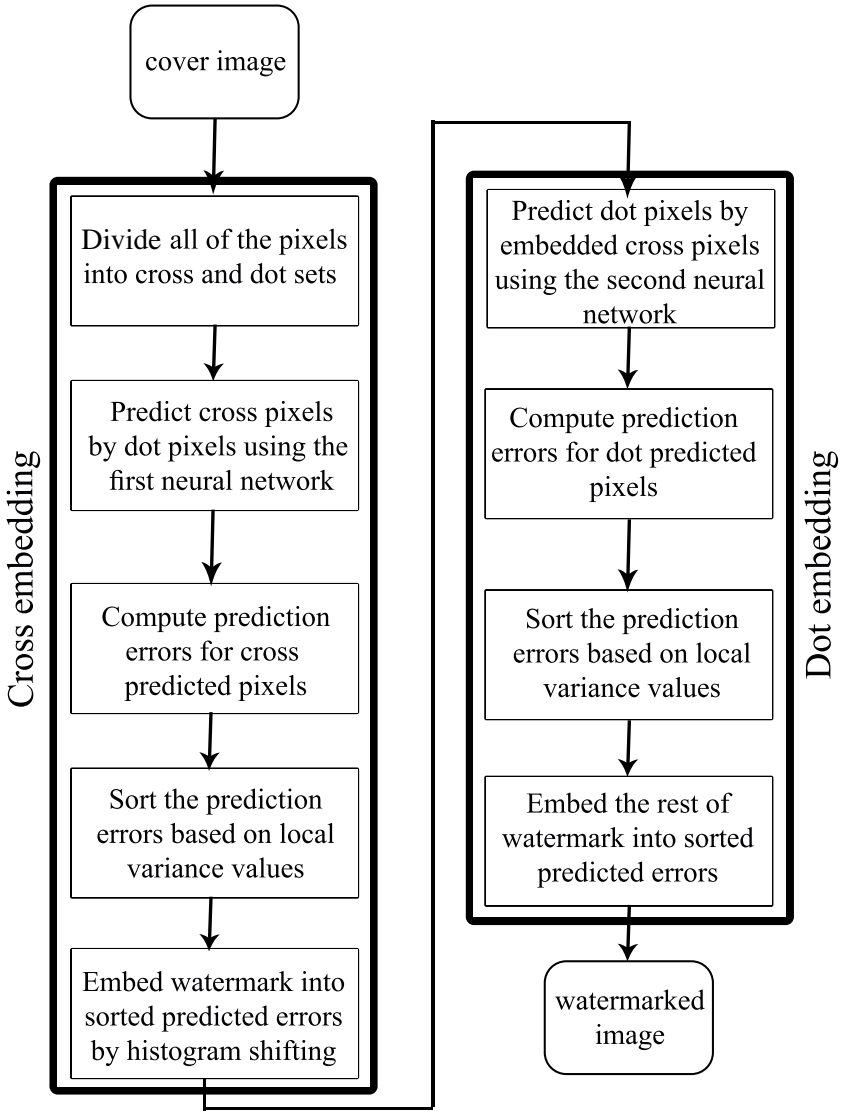

Figure 5. Proposed watermark-embedding procedure.

3.2. Proposed extracting and recovery procedures

The extracting and recovery procedures are inverse in the embedding procedure. The cross-extracting and recovery procedures are as follows:

Find all the cells whose central pixel belongs to the cross set.

Predict the cross pixel in the center of each cell using the first neural network.

Compute the prediction errors $d_{i,j}$ by Eq. (8).

Compute the local variance values for all cells by Eq. (14).

Sort the cells based on their local variance values.

Read the first 34 LSBs from the sorted cells and restore the values of T_n and T_p and the length of watermark that is embedded in the image.

Skip the first 34 cells of the sorted cells.

Extract the watermark as presented by Sachnev et al. [4].

Recover the original LSBs from the first 34 sorted prediction errors.

Restore the original pixel values $u_{i,j}$ with Eq. (13).

4. Experimental results

In this paper, we use two artificial neural networks for predicting pixel values in reversible watermarking. We tested a different number of hidden layers and a different number of neurons in the hidden layers, and we selected the architecture that produces maximum accuracy for the neural network. We obtained one hidden layer and 18 neurons in each hidden layer for both neural networks. In this paper, for the purpose of training the neural networks, we used the error backpropagation method and the sigmoid function as the activation function. The sigmoid function was computed by Eq. (17).

$$\varphi(x) = \frac{1}{1+e^{-x}} \tag{17}$$

The initial weight values are defined randomly in the range of [–1, 1], and the learning rate α is equal to 0.15. We stopped training neural networks using a threshold value equal to 0.001. Obviously, smaller threshold values lead to increased training time and convergence of the neural network. Larger threshold values may reduce the computational complexity and convergence time of the network but lead to poor predictions and an inefficient embedding operation, resulting in smaller PSNR values. Our proposed method is compared to several methods proposed by Sachnev et al. [4], Thodi and Rodriguez [5], Jung et al. [9], Al Qershi and Khoo [11], and other methods [2,6–8,12–15]. The results are shown for four 512 × 512 eight-bit gray-scale test images: Lena, Baboon, Barbara, and Airplane. These images are shown in Figure 6.

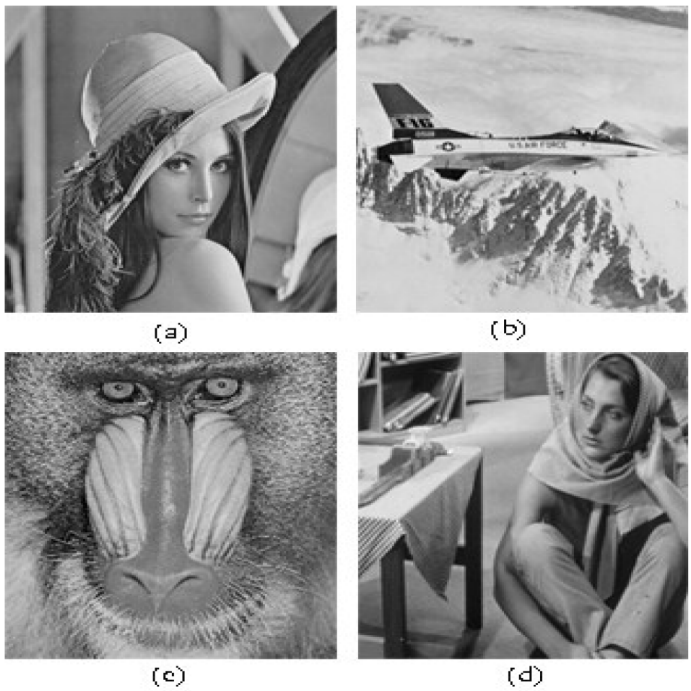

Figure 6. Four standard 512 × 512 gray-scale test images: (a) Lena, (b) Baboon, (c) Airplane, (d) Barbara.

The PSNR criterion is used to evaluate the quality of the watermarked image. For a M × N gray-scale image, the PSNR value is described by Eq. (18).

$$PSNR = 10 \times \log_{10}^{\frac{255^2}{MSE}} \tag{18}$$

The mean squared error (MSE) criterion in Eq. (18) is used to quantify the difference between the original and the watermarked image. The MSE value is described by Eq. (19).

$$MSE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(c_{i,j} - s_{i,j})^2}{M \times N} \tag{19}$$

In Eq. (19), $c_{i,j}$ and $s_{i,j}$ are the pixels of the original image and the watermarked image, respectively, whose coordinates are (i,j).

In Figure 7, capacity-versus-distortion results are shown for four test images. Among the four experimental images, the Airplane image is the smoothest image and the Baboon image has the maximum number of edges. A smooth image makes an accurate prediction and has small prediction errors, and therefore it is an efficient embedding procedure. For this reason, as shown in Figure 7, the Airplane image has maximum efficiency and the Baboon image has minimum efficiency in the embedding procedure. In Figure 8, the effect of the different threshold values in dB is shown in the quality of the embedded image. We compare the effect of using different threshold values on PSNR values for threshold values Tn and Tp equal to [–1, 0], [–1, 1], [–3, 3], and [–8, 7]. The smaller threshold values resulted in larger PSNR values in the equal size of the embedded watermark, and the larger threshold values resulted in minimum PSNR values.

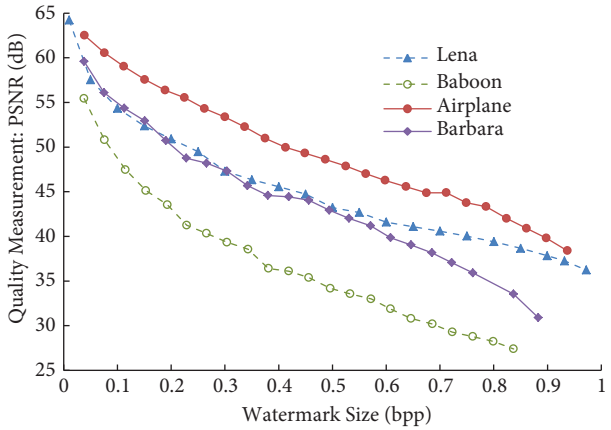

Figure 7. Quality-versus-capacity results for four test images.

In Table 1, we compare the values of the watermarked image quality by using the PSNR criterion between our proposed method and many other methods [2,4–9,11–15] in some hiding capacities from 0.1 to 0.9 bpp for the Lena image. As this table shows, our proposed scheme has higher PSNR values in all hiding capacities for the Lena image. According to Figure 8, we used the appropriate threshold values to embed the watermark in the image. In the embedding procedure, in order to embed payloads of less than 0.3 bpp, we used negative and positive threshold values equal to –1 and 0, respectively. For embedding payloads between 0.3 and 0.4 bpp, we used negative and positive threshold values equal to –1 and 1, respectively. For embedding payloads between 0.4

and 0.7 bpp, we used negative and positive threshold values equal to –3 and 3, respectively, and for embedding larger payloads, we used negative and positive threshold values equal to –8 and 7, respectively. Using these appropriate threshold values makes for an efficient embedding procedure and results in higher PSNR values. In Table 1, "-" means that the mentioned method cannot embed a watermark of that size in an image.

Table 1. PSNR comparison results for the proposed scheme and other methods for the Lena image in dB.

0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	Payload (bpp)
32.1	33.2	34.4	35.4	36.8	39.5	41.8	43.4	46.2	Tian [2]
32.6	34.1	34.9	36	36.8	37.6	38.4	39.1	39.7	Thodi [5]
36.5	38.2	39.7	41.1	42.6	44.8	46.8	50.6	54	Sachnev [4]
-	-	35	32.8	34.4	35.8	37.5	39.9	43.5	Celik [6]
-	-	-	-	36.8	38.2	40.1	41.9	44.8	Yang [7]
-	-	-	37.3	37.9	38.8	40.3	42.6	45.3	Hsiao [8]
-	32	33.1	35	36.8	38.9	40.9	43.7	47.1	Jung [9]
-	-	-	-	42.8	43.9	45.6	47.3	51.8	Al-Qershi [11]
-	34.7	36.7	38.9	41.3	43.4	46.5	-	-	Luo [12]
34.4	36.1	37.6	38.9	40.7	42.1	44.2	48.4	51.6	Hu [13]
-	-	33	35.1	36.9	40.3	43.2	47.1	-	Tai [14]
32.6	34.1	35.5	37.6	39.9	42.2	45	48.4	-	Tsai [15]
37.9	39.5	40.7	41.8	43.4	45.7	47.4	51	54.5	Proposed

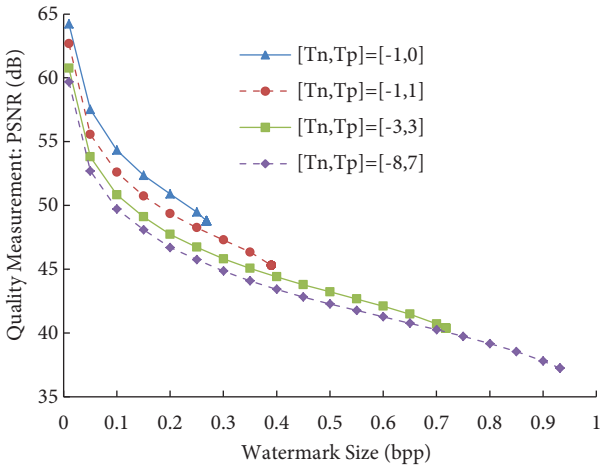

Figure 8. Effect of using different threshold values on image quality in dB for the Lena image.

In Table 2, hiding capacity and PSNR values for different threshold values are shown for the test images. Table 2 shows that larger threshold values cause larger embedding capacities and smaller PSNR values. Larger threshold values cause more prediction errors and can be used for embedding the watermark. This results in greater capacity and smaller PSNR values.

Table 2. Hiding capacity and PSNR values for the proposed method in different threshold values for four images.

PSNR (dB)				Embedding capacity (bpp)				Image
[–12,12]	[–8,7]	[–3,3]	[–1,0]	[–12,12]	[–8,7]	[–3,3]	[–1,0]	
36.2311	37.2454	40.3743	48.7958	0.9722	0.9319	0.7178	0.2683	Lena
34.0914	35.7491	39.9106	48.7627	0.9355	0.8599	0.6345	0.2543	Baboon
40.4013	41.5578	44.6217	52.1156	0.9736	0.9426	0.8127	0.3859	Airplane
33.2024	35.2054	39.5832	48.6539	0.8987	0.8227	0.6005	0.2113	Barbara

In Table 3, maximum hiding capacity is compared to that of Sachnev et al. [4] for different threshold values. As this table shows, our proposed scheme achieves higher embedding capacities compared to Sachnev et al. for all tested images and threshold values.

Table 3. Maximum hiding capacity for the proposed scheme and that of Sachnev et al. in bpp.

[Tn, Tp]	Lena		Baboon		Airplane		Barbara	
	Sachnev et al. [4]	Proposed method	Sachnev et al. [4]	Proposed method	Sachnev et al. [4]	Proposed method	Sachnev et al. [4]	Proposed method
[−1,0]	0.2456	0.2683	0.2222	0.2543	0.3166	0.3859	0.1844	0.2113
[−2,1]	0.4585	0.4909	0.3974	0.4464	0.5770	0.6354	0.3479	0.3969
[−3,3]	0.6941	0.7178	0.5800	0.6345	0.7800	0.8127	0.5370	0.6005
[−5,5]	0.8373	0.8625	0.7141	0.7709	0.8658	0.8979	0.6639	0.7389
[−6,6]	0.8748	0.8997	0.7606	0.8142	0.8900	0.9204	0.7022	0.7798
[−8,7]	0.9090	0.9319	0.8126	0.8599	0.9154	0.9426	0.7424	0.8227
[−12,12]	0.9592	0.9722	0.9077	0.9355	0.9587	0.9736	0.8181	0.8987

In Figure 9, comparison results of distortion versus capacity are shown for our proposed scheme, as well as three other methods presented by Sachnev et al. [4], Luo et al. [12], and Hu et al. [13]. As seen in Figure 9, our proposed scheme achieves higher PSNR values at all embedding capacities and for all test images. In all the images, our proposed scheme is the top curve. In Figures 10 and 11, the embedding results for the Lena and Baboon images at different threshold values are shown. We set the T_n and T_p thresholds to [–1, 0], [–5, 5], and [–12, 12] values. Larger threshold values resulted in larger capacity values and smaller PSNR values. In Figure 10, maximum threshold values are set to [–12, 12]. For the Lena image these threshold values resulted in 0.97 bpp capacity with 36.23 dB. Hence, the embedded images are not easily detectable from the original image by the human visual system.

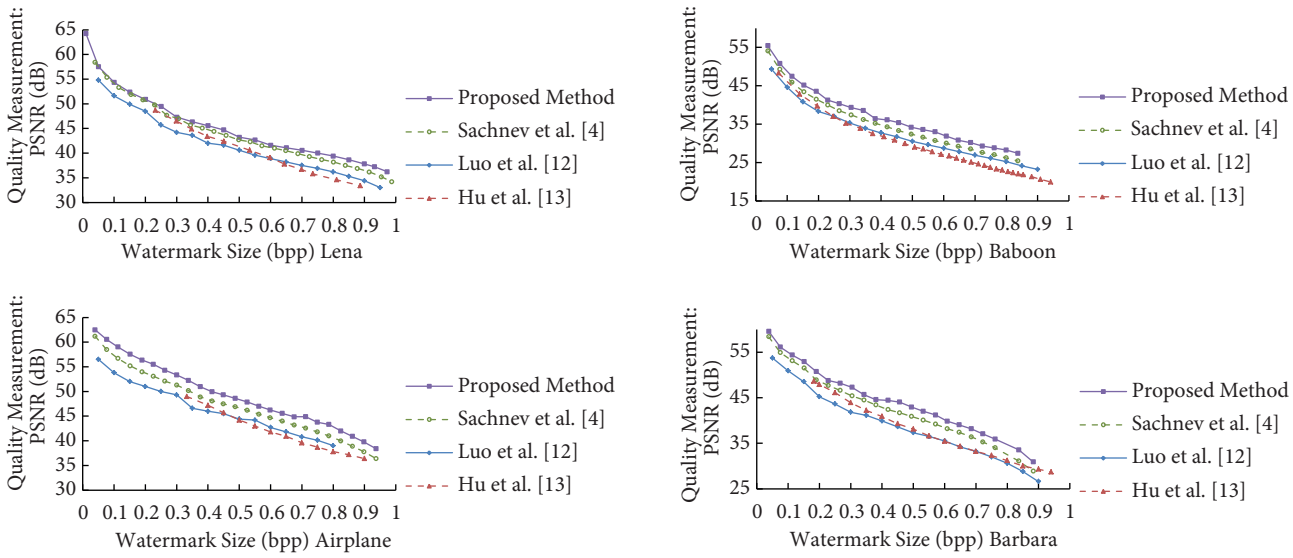

Figure 9. Comparison results of distortion versus capacity for our proposed scheme and other methods on four images.

Figure 10. Lena-embedding results: (a) original, (b) 48.79 dB embedded with 0.27 bpp with $[T_n\ ,\ T_p] = [–1,0]$, (c) 38.41 dB embedded with 0.86 bpp with $[T_n\ ,\ T_p] = [–5,5]$, (d) 36.23 dB embedded with 0.97 bpp with $[T_n\ ,\ T_p] = [–12,12]$.

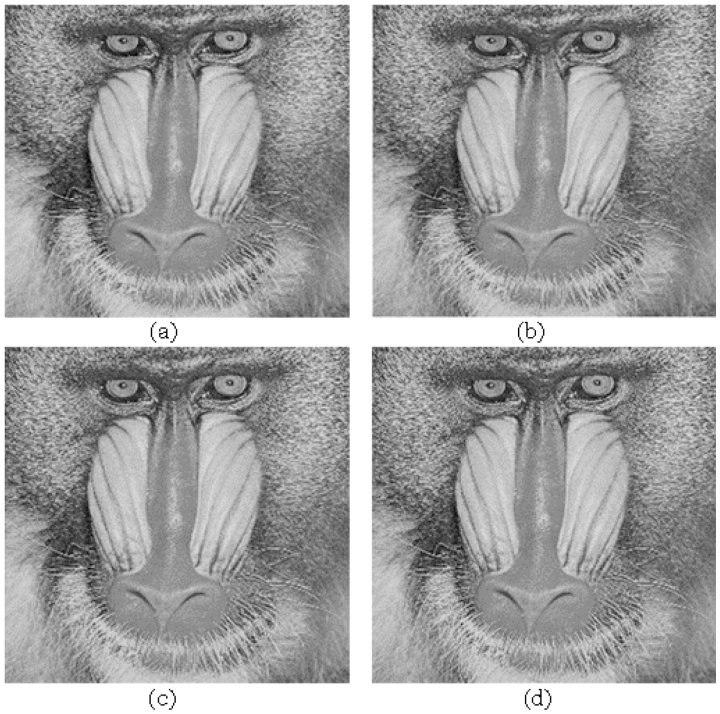

Figure 11. Baboon-embedding results: (a) original, (b) 48.76 dB embedded with 0.25 bpp with $[T_n\ ,\ T_p] = [–1,0]$, (c) 37.45 dB embedded with 0.77 bpp with $[T_n\ ,\ T_p] = [–5,5]$, (d) 34.09 dB embedded with 0.93 bpp with $[T_n\ ,\ T_p] = [–12,12]$.

5. Conclusion

In this paper, an intelligent approach for predicting pixel values was proposed using artificial neural networks. Artificial neural networks are powerful tools of prediction. We used the neural networks in a rhombus pattern technique to predict the pixel values instead of averaging neighboring pixels. In many cases, the averaging technique did not offer accurate predictions. For example, in the nonsmooth areas of the image, the averaging technique was not accurate for prediction, yet the proposed scheme had a high predicting power. The artificial neural networks provided an accurate prediction for all areas of the image, including nonsmooth areas. This technique provided an accurate prediction, and, consequently, smaller prediction errors were generated. Since watermark bits were embedded in the prediction errors, using artificial neural networks resulted in a better embedding procedure that reduced the image distortion after watermark embedding. The experimental results were compared with the proposed methods presented by Sachnev et al. [4], Thodi and Rodriguez [5], Jung et al. [9], Al-Qershi and Khoo [11], and other methods [2,6–8,12–15]. The superiority of the proposed method was demonstrated in the experimental results.

References

[1] Lee CD, Kalker T. Reversible image watermarking based on integer-to-integer wavelet transform. IEEE T Inf Foren Sec 2007; 2: 321-330.

[2] Tian J. Reversible data embedding using a difference expansion. IEEE T Circ Syst Vid 2003; 13: 890-896.

[3] Ni Z, Shi YQ, Ansari N, Su W. Reversible data hiding. IEEE T Circ Syst Vid 2006; 16: 354-362.

[4] Sachnev V, Kim HJ, Nam J, Suresh S, Shi YQ. Reversible watermarking algorithm using sorting and prediction. IEEE Trans Circ Syst Vid 2009; 19: 989-999.

[5] Thodi DM, Rodriguez JJ. Expansion embedding techniques for reversible watermarking. IEEE T Image Process 2007; 16: 721-730.

[6] Celik MU, Sharma G, Tekalp AM, Saber E. Reversible data hiding. In: ICIP 2002 International Conference on Image Processing; 22–25 September 2002; Rochester, NY, USA. New York, NY, USA: IEEE. pp. 157-160.

[7] Yang B, Schmucker M, Funk W, Busch C, Sun S. Integer DCT-based reversible watermarking technique for images using companding technique. J Electron Imaging 2004; 5306: 405-415.

[8] Hsiao J, Chan K, Chang J. Block-based reversible data embedding. Signal Process 2009; 89: 556-569.

[9] Jung S, Ha LT, Ko S. A new histogram modification-based reversible data hiding algorithm considering the human visual system. IEEE Signal Proc Let 2011; 18: 95-98.

[10] Al-Qershi OM, Khoo BE. Reversible watermarking scheme based on two-dimensional difference expansion (2D-DE). In: IEEE 2010 International Conference on Computer Research and Development; 7–10 May 2010; Kuala Lumpur, Malaysia. New York, NY, USA: IEEE. pp. 228-232.

[11] Al-Qershi OM, Khoo BE. Two-dimensional difference expansion (2D-DE) scheme with a characteristics-based threshold. Signal Process 2013; 93: 154-162.

[12] Luo L, Chen Z, Chen M, Zeng X, Xiong Z. Reversible image watermarking using interpolation technique. IEEE T Inf Foren Sec 2010; 5: 187-193.

[13] Hu Y, Lee HK, Li J. DE-based reversible data hiding with improved overflow location map. IEEE T Circ Syst Vid 2009; 19: 250-260.

[14] Tai WL, Yeh CM, Chang CC. Reversible data hiding based on histogram modification of pixel differences. IEEE T Circ Syst Vid 2009; 19: 906-910.

[15] Tsai P, Hu YC, Yeh HL. Reversible image hiding scheme using predictive coding and histogram shifting. Signal Process 2009; 89: 1129-1143.

[16] Tian J. Wavelet-based reversible watermarking for authentication. J Electron Imaging 2002; 4675: 679-690.

[17] Tian J. High capacity reversible data embedding and content authentication. In: IEEE 2003 International Conference on Acoustics, Speech, and Signal Processing; 6–10 April 2003; Hong Kong. New York, NY, USA: IEEE. pp. 517-520.

[18] Tian J, Wells RO. Reversible data-embedding with a hierarchical structure. In: ICIP 2004 International Conference on Image Processing; 24–27 October 2004; Singapore. New York, NY, USA: IEEE. pp. 3419-3422.

[19] Alattar M. Reversible watermark using difference expansion of triplets. In: ICIP 2003 International Conference on Image Processing; 14–17 September 2003; Barcelona, Spain. New York, NY, USA: IEEE. pp. 501-504.

[20] Alattar M. Reversible watermark using difference expansion of quads. In: IEEE 2004 International Conference on Acoustics, Speech, and Signal Processing; 17–21 May 2004; Montreal, QC, Canada. New York, NY, USA: IEEE. pp. 377-380.

[21] Alattar M. Reversible watermark using the difference expansion of a generalized integer transform. IEEE T Image Process 2004; 13: 1147-1156.

[22] Maity HK, Maity SP. Intelligent modified difference expansion for reversible watermarking. Int J Multimed Appl 2012; 4: 83-95.

[23] Maity HK, Maity SP, Maity D. Modified difference expansion for reversible watermarking using fuzzy logic based distortion control. Adv Comput Inf Technol 2013; 177: 871-883.

[24] Lin CC, Yang SP, Hsueh NL. Lossless data hiding based on difference expansion without a location map. In: IEEE 2008 International Conference on Image and Signal Processing; 27–30 May 2008; Sanya, China. New York, NY, USA: IEEE. pp. 8-12.

[25] Kamstra LHJ, Heijmans AM. Reversible data embedding into images using wavelet technique and sorting. IEEE T Image Process 2005; 14: 2082-2090.

[26] Kim HJ, Sachnev V, Shi YQ, Nam J, Choo HG. A novel difference expansion transform for reversible data embedding. IEEE T Inf Foren Sec 2008; 3: 456-465.

[27] Vleeschouwer CD, Delaigle JE, Macq B. Circular interpretation of histogram for reversible watermarking. In: IEEE 2001 Workshop on Multimedia Signal Processing; 3–5 October 2001; Cannes, France. New York, NY, USA: IEEE. pp. 345-350.

[28] Yang B, Schmucker M, Niu X, Busch C, Sun S. Reversible image watermarking by histogram modification for integer DCT coefficients. In: IEEE 2004 Workshop on Multimedial Signal Processing; 29 September–1 October 2004; Siena, Italy. New York, NY, USA: IEEE. pp. 143-146.

[29] Thodi DM, Rodriguez JJ. Reversible watermarking by prediction-error expansion. In: IEEE 2004 Southwest Symposium on Image Analysis and Interpretation; 28–30 March 2004; Lake Tahoe, NV, USA. New York, NY, USA: IEEE. pp. 21-25.

[30] Hong W, Chen TS, Shiu CW. Reversible data hiding for high quality images using modification of prediction errors. J Syst Software 2009; 82: 1833-1842.

[31] Kang SU, Hwang HJ, Kim HJ. Reversible watermark using an accurate predictor and sorter based on payload balancing. ETRI J 2012; 34: 410-420.

[32] Kotvicha A, Sanguansat P, Kulthon Kasema ML. Expand variance mean sorting for reversible watermarking. Int J Comput Commun 2012; 1: 196-199.

[33] Hong W, Chen TS, Shiu CW. Reversible data hiding for high quality images using modification of prediction errors. J Syst Software 2009; 82: 1833-1842.

[34] Kang SU, Hwang HJ, Kim HJ. Reversible watermark using an accurate predictor and sorter based on payload balancing. ETRI J 2012; 34: 410-420.

[35] Kim KS, Lee MJ, Lee HY, Lee HK. Reversible data hiding exploiting spatial correlation between sub-sampled images. Pattern Recogn 2009; 42: 3083-3096.

[36] Fausett L. Fundamentals of Neural Networks Architectures, Algorithms and Applications. London, UK: Pearson Education, 1993.