*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105 (Online)
Volume **9**, Number **2**, May **2024**

# Reversible Watermarking Based on Wavelet Packet Transform and Deep Learning-Diamond Predictor

Xiuyan Sun

Public Education Department
Laiwu Vocational and Technical College
sunxiuyan0634@163.com

Chengling Gu

Academic Affairs Office
Laiwu Vocational and Technical College
chengling_2119@163.com

Chuanxing Li

Mathematics Teaching
Jinan Laiwu Chenyi Middle School
13506343390@163.com

Jianchuan He

Harbin Institute of Technology, Shenzhen
863646236@qq.com

Linlin Tang*

Harbin Institute of Technology, Shenzhen
hittang@126.cm

*Corresponding author: Linlin Tang

Received August 2, 2023, revised November 20, 2023, accepted February 6, 2024.

Abstract. *Wavelet analysis is a powerful tool for time-frequency analysis. Wavelet packet analysis divides the time-frequency plane more carefully. Its resolution of the high-frequency part of the signal is higher than that of dyadic wavelet. Watermarking scheme based on deep learning-diamond predictor and wavelet packet transform can achieve greater embedding capacity and less image distortion than other methods, so it has been researched in recent years. A novel digital watermark algorithm is proposed in this paper. Good experiments show its effectiveness.*
**Keywords:** Reversible Watermarking, Wavelet Packet Transform, Ridgelet Transform, Deep Learning-Diamond Predictor

1. **Introduction.** According to different domains, digital watermarking embedding techniques are mainly divided into three categories: time/space domain algorithm, transform domain algorithm and compression domain algorithm [1]. Reversible watermarking [2] is a special digital watermarking technology. According to the degree of impact of embedded information on the carrier, it can be divided into reversible watermarks and irreversible watermarks. Reversible watermarking restores the carrier to the state before embedding the watermark after extracting the watermark information, without causing permanent interference to the carrier. Irreversible watermarks cannot restore the carrier to its pre

embedding state after extracting the watermark, and the impact of embedding the watermark on the carrier is permanent.

As a mathematical tool, wavelet transform is a well-known Fourier transform and window Fourier transform. It has an important impact on signal analysis. Wavelet analysis theory developed from multi-scale analysis, time-frequency analysis and pyramid algorithm has become the most useful tool for processing and analysis.

After two-dimensional wavelet transform, the original image is decomposed into sub images with different scales. After one-level wavelet decomposition, the original image can be divided into four frequency bands: the low-frequency sub image LL that retains most of the information of the original image, the high-frequency horizontal sub image (LH), the high-frequency vertical sub image (HL) and the high-frequency diagonal sub image (HH) that contain high-frequency information such as edge details and textures. Next level of wavelet decomposition only further decomposes the low-frequency subgraph LL of the image to obtain low-frequency components and high-frequency components, wavelet packet decomposition not only further decomposes low-frequency part (LL), but also further decomposes the high-frequency part (LH, HL, HH) of the image to more carefully depict the high-frequency part of the signal, so as to have a stronger ability to analyze the signal. Therefore, four subgraphs are decomposed by two-level wavelet packet decomposition, and 16 subgraphs can be obtained. By analogy, 64 subgraphs can be obtained after the third level decomposition. It greatly expands the space of embedded information, because the amount of embedded signal is proportional to robustness of watermark, so it further improves the robustness of the watermark.

Early prediction error expansion techniques required a significant waste of space to store Bitmaps, and due to low prediction accuracy, prediction errors were generally large, resulting in significant distortion of watermark images. Diamond prediction techniques which was proposed by Sachnev et al. It has been greatly improved prediction accuracy by dividing the image into chessboards and using half of pixel values to predict other, thereby significantly increasing capacity of watermark embedding. From then on, many improvements and optimizations based on diamond predictors have emerged.

A new deep neural network-based [3, 4, 5] predictor based on the diamond predictor, cleverly combining deep learning [6, 7] with traditional reversible watermarking schemes has been proposed to design a reversible watermarking algorithm with good performance [8, 9, 10]. Related work will be given in section 2. Our proposed method will be shown in section 3 and the experimental results will be introduced in section 4.

## 2. **Related Work.**

### 2.1. **Wavelet Packet Decomposition.**

Wavelet packet analysis can provide a more precise analysis method for signals.

Let $\varphi(x)$ and $\psi(x)$ be the scaling function and wavelet function, let

$$\begin{cases} \psi_0(x) = \varphi(x) \\ \psi_1(x) = \psi(x) \\ \varphi_{2l}(x) = \sum_{k=-\infty}^{+\infty} h_k \varphi_l(2x-k) \\ \psi_{2l+1}(x) = \sum_{k=-\infty}^{+\infty} g_k \varphi_l(2x-k) \end{cases} \tag{1}$$

Then, the defined function $\{\psi_n\}$ is called wavelet packet about scaling function $\varphi(x)$. Let $\varphi(x)$ and $\psi(x)$ be the scaling function and wavelet function, let

$$\begin{cases} \psi_0(x) = \varphi(x) \\ \psi_1(x) = \psi(x) \\ \varphi_{2^l}(x) = \sum_{k=-\infty}^{+\infty} h_k \varphi_{2^{l-1}}(2x-k) \\ \psi_{2^l+1}(x) = \sum_{k=-\infty}^{+\infty} g_k \varphi_{2^{l-1}}(2x-k) \end{cases} \tag{2}$$

Then, defined function $\{\psi_n\}$ is called the shortened wavelet packet about scaling function $\varphi(x)$.

2.2. **Diamond Predictor.** The first step of diamond predictor prediction is to divide the image into disjoint point sets and cross sets, as shown in Figure 1. The point set pixels remain unchanged, and the cross set pixel values are predicted using the point set pixels. Then, the cross set pixel values are modified by expanding the prediction error to embed the watermark. Due to fact that all the pixels in the point set have not changed before and after embedding the watermark, it is possible to ensure that the predicted values of the cross set pixels remain unchanged during watermark extraction.

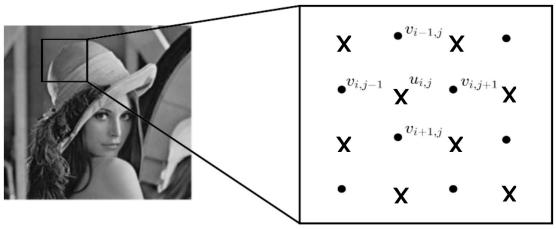



FIGURE 1. Diamond Predictor

For the predicted cross set pixel $a$, its predicted value $b$ is obtained by averaging the pixel values of the surrounding four point sets, as shown in Formula (3).

$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor \tag{3}$$

Based on the true value $u_{i,j}$ and the predicted value $u'_{i,j}$, the prediction error $e_{i,j} = u_{i,j} - u'_{i,j}$ can be obtained. And through extending $e_{i,j}$, watermark embedding can be achieved. Sachnev et al. [11] proposed a new prediction error modification scheme, which limits the threshold parameters of the prediction error extension and reduces the distortion of the watermark image. Design two thresholds for prediction error $T_n < 0$, $T_p \geq 0$ then expand and embed the pixels with prediction errors $e_{i,j} \in [T_n, T_p]$, and shift the points with prediction errors $e_{i,j} \notin [T_n, T_p]$, as shown in Formula (4).

$$e'_{i,j} = \begin{cases} 2e_{i,j} + b, \text{ if } e_{i,j} \in [T_n, T_p] \\ e_{i,j} + T_p + 1, \text{ if } e_{i,j} > T_p \text{ and } T_p \geq 0 \\ e_{i,j} + T_n, \text{ if } e_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \tag{4}$$

Taking $T_n = -1$ and $T_p = 0$ as examples, this modification method is equivalent to shifting all points with prediction error $e_{i,j} < -1$ to the left, leaving $-2$ positions for embedding watermark information, shifting all points with prediction error values $e_{i,j} > 0$ to the right, and leaving 1 position for embedding watermark information, as shown in Figure 2.

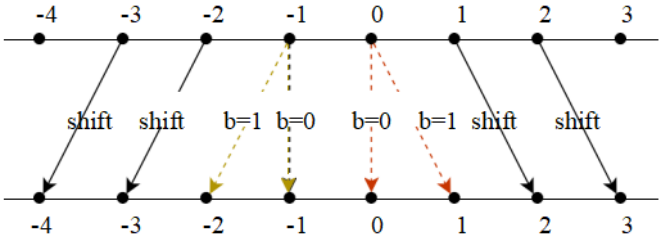



FIGURE 2. Modification of Prediction Error

3. **Proposed Method.** A new deep neural network-based predictor is proposed based on the diamond predictor, which cleverly combines deep learning with traditional reversible watermarking schemes. Since the diamond prediction method preserves half of the original pixel points, when embedding and extracting, if only this half of the unchanged pixel points are input into the neural network, a fixed and unchanged output result can be obtained. Afterwards, watermark is embedded by modifying the values of the predicted pixels. By using this predictor, the overall prediction error is greatly reduced compared to traditional methods, and there are more prediction errors available for expanding and embedding watermark information during watermark embedding, resulting in a smaller number of invalid shift prediction errors. In this paper, main work is focused on the selection of wavelet coefficients for watermark embedding based on the decomposition.

3.1. **Blocking and Transforming.** Original image will firstly be blocked according to original content. Main purpose is to distinguish the smoothness and complexity part. Large blocks are used where the details of image smoothing are not rich, and smaller blocks are used where the details are rich. Then, different transform can be considered used on them. So the next step is to find more suitable positions for embedding watermark in the wavelet domain after the subsequent transformation. The Figure 3 below shows an example of this blocking.

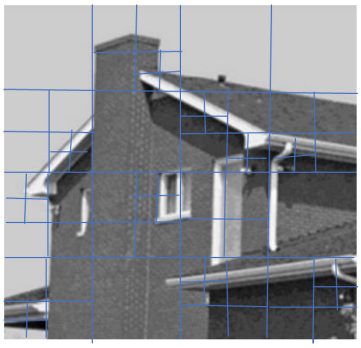


FIGURE 3. Example of Blocking According to Image Characteristics

After blocking process, we firstly determine which blocks are used for wavelet packet transform. For larger blocks, edge information and details are not so rich, so wavelet packet transform is selected. For small blocks with rich edge and detail information, we select ridgelet transform to obtain better edge details.

Wavelet decomposition information with both spatial and frequency domain characteristics, and from the decomposition results, rich detailed information about the original image can be obtained. After the above blocking operation, the information obtained through wavelet packet or ridge wave decomposition can be seen as a refinement of local

region information at different scales. This provides more selectable regions for watermark embedding. One of our tasks is to combine the subsequent watermark embedding process to filter out good embedding blocks to achieve better robustness.

3.2. **Predictor Training.** Thanks for vigorous development of deep learning in the field of image denoising, the prediction task of the reversible watermark scheme proposed in this paper is related to the image denoising task. Therefore, the neural network model used in this paper is the SGN network (Self Guided Network) originally applied in the field of image denoising. The overall structure of the network model is shown in Figure 4, but the selection of the model is not fixed. Many similar models can achieve significant results, such as U-Net. The dataset used for training is 90000 randomly selected images from ImageNet.

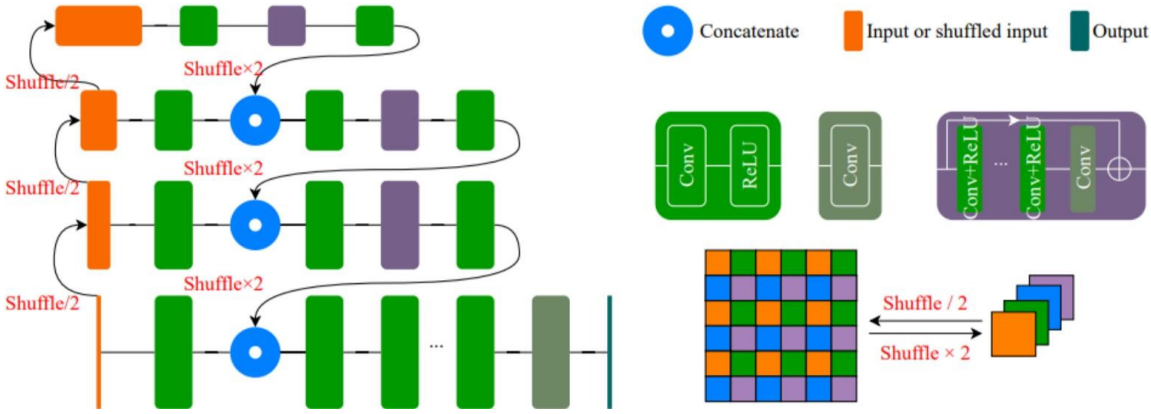



FIGURE 4. Overall Structure of SGN Network Model

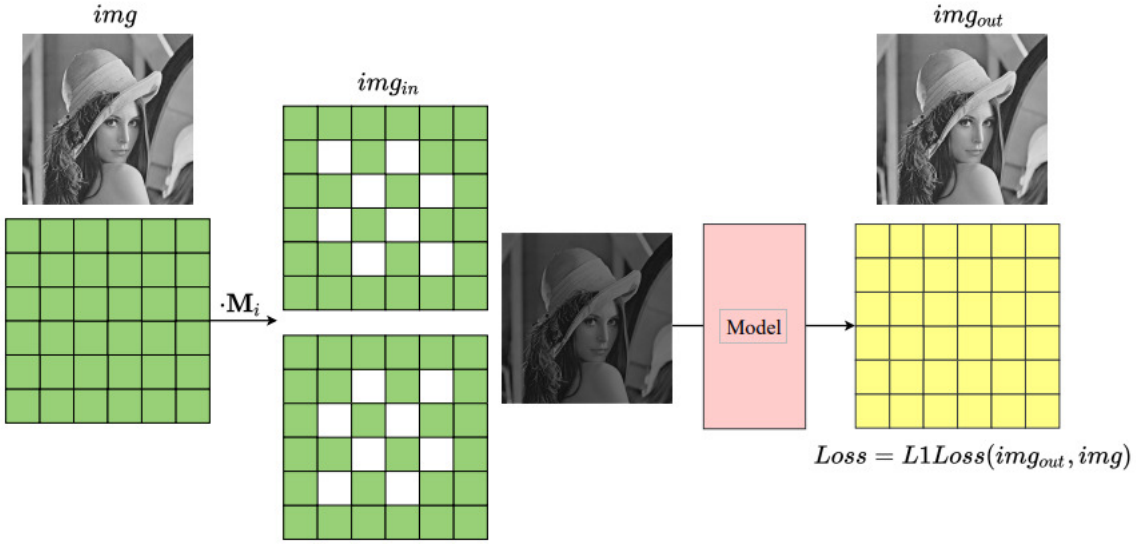



FIGURE 5. Training Method of Model

The training method of the model is shown in Figure 5. First, the pictures in the training set are randomly scaled and rotated, and then cut to a fixed size. After being converted to a grayscale, the labels $img$ of the original image, $M1$ and $M2$ are two binary matrices of the same size used to divide the picture into chessboards, which meet the conditions of Equations (5) and (6). In fact, it is equivalent to dividing all binary

matrices into one except the outermost circle. Set the remaining pixels to disjoint 0 and 1 values, as shown in Figure 7.

$$M_1[i][j] = \begin{cases} 1, \text{ if } i,j = 0 \text{ or } 511 \\ (i+j) \bmod 2, \text{ else} \end{cases} \tag{5}$$

$$M_2[i][j] = \begin{cases} 1, \text{ if } i,j = 0 \text{ or } 511 \\ (i+j+1) \bmod 2, \text{ else} \end{cases} \tag{6}$$

Then dot multiply $img$ by a random $M_i$ $(i = 1, 2)$(equivalent to adding noisy to the


FIGURE 6. Schematic Diagram of $M_1$ Matrix and $M_2$ Matrix

image), which obtains input of model $img_{in}$. According to research of Zhao et al. [12], the performance of $L_1$ Loss in the image task is obviously better than that of $L_2$ Loss, so the loss function used by the network is $L_1$ Loss.

$$L^{l_1} = \frac{1}{N} \sum_{p \in P} |img_{out}(p) - img(p)| \tag{7}$$

Where $P$ set represents the set of position $(i, j)$ which satisfy $M_i(i, j) = 0$.

3.3. **Watermark Embedding.** Before embedding watermark, it is necessary to preprocess image and use the $LSB$ replacement method to extract the $LSB$ bit in the zero line of the image after wavelet transform decomposition, which is then embedded as part of the watermark into the image. $LSB$ bit in zero line is used to embed the key parameter information necessary for extracting the watermark and restoring the image. Parameters include: number of embedding layers used to store the watermark, level parameter, prediction error threshold parameter selected for embedding watermark in that layer, length of the watermark, and then the remaining bits are used to store the length of the overflow bit after bitmap compression. The main embedding steps of the watermark are shown in Figure 7.

Multiply preprocessed image $img$ and matrix $M_1$ to obtain input of model: $img_{in1} = ing \cdot M_1$. Input $img_{in1}$ into the model for prediction to obtain the output image $img_{out1}$, and then modify $img_{out1}$. Mark the position set of all 1 elements in $M_1$ as $P^{(1)}_{M_1}$, and the position set of all 0 elements as $P^{(0)}_{M_1}$, that is:

$$\begin{aligned} P^{(1)}_{M_1} &= \{(x, y) \mid M_1(x, y) = 1\} \\ P^{(0)}_{M_1} &= \{(x, y) \mid M_1(x, y) = 0\} \end{aligned} \tag{8}$$

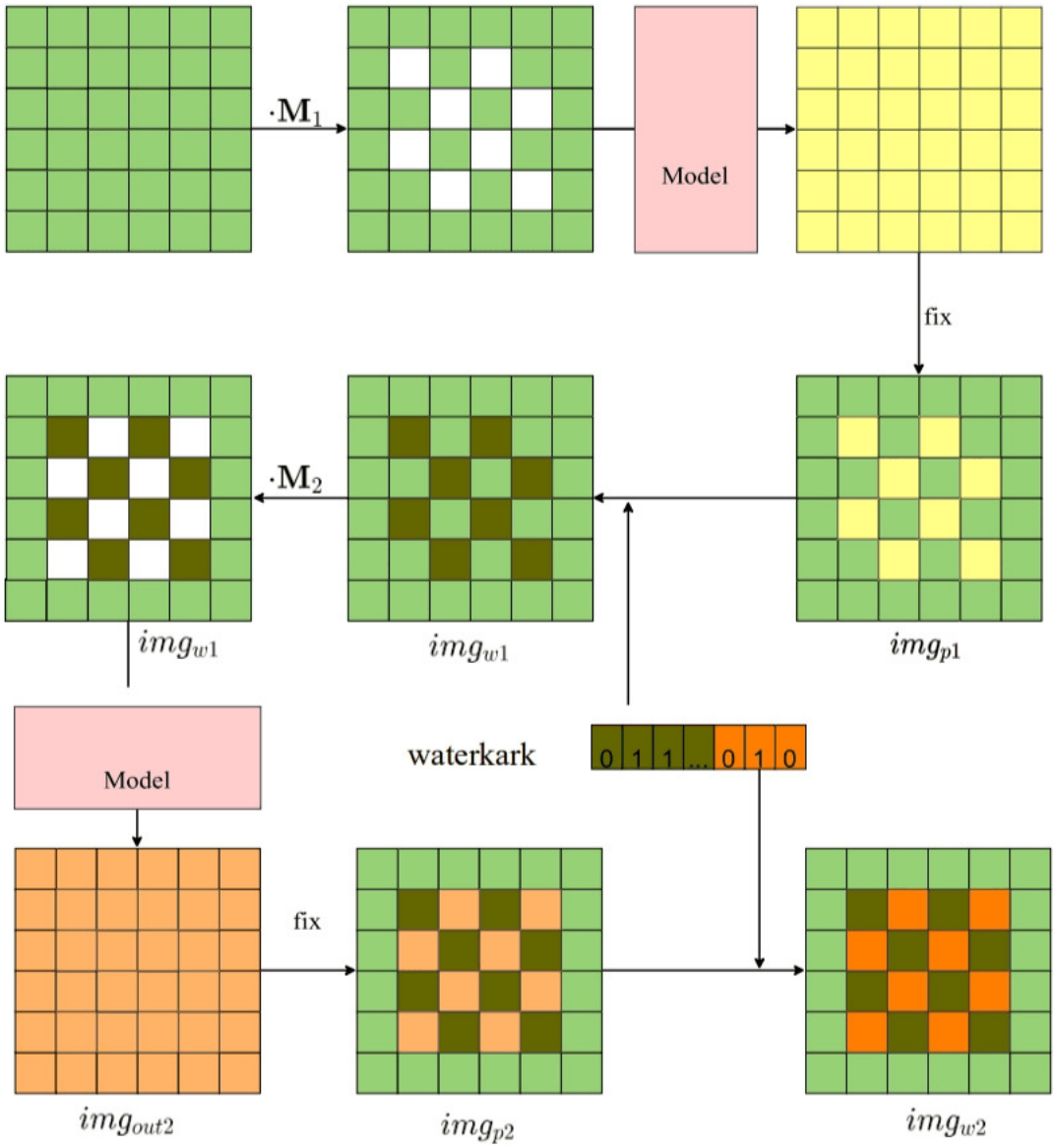



FIGURE 7. Watermark Embed Scheme

Then modify all the pixel values at position $P_{M_1}^{(1)}$ in $img_{out1}$ to those in the original $img$ image to obtain the predicted image $img_{p1}$, as shown in Formula (9), where $\overline{M_1}$ represents the inverse of the matrix $M_1$, changing values of all elements at position $P_{M_1}^{(0)}$ to 1 and all elements at position $P_{M_1}^{(1)}$ to 0.

$$img_{p1} = img_{\text{out1}} \cdot \overline{M_1} + img \cdot M_1 \tag{9}$$

Subtract the grayscale values of the position pixels corresponding to $P_{M_1}^{(0)}$ in $img$ and $img_{p1}$ to generate a prediction error:

$$PE(x,y) = img(x,y) - img_{p1}(x,y), \quad (x,y) \in P_{M_1}^{(0)} \tag{10}$$

Define $Neighbour(x,y) = \{(i,j) \mid |i-x|+|j-y| = 1\}$ as neighbor pixel set of pixel point $(x,y)$, use $\overline{Neighbour(x,y)}$ to represent the average value of all neighbor pixels of $(x,y)$, design local complexity function $\rho(x,y)$ as shown in Formula (11), where $|Neighbour(x,y)|$ represents the number of elements of set $Neighbour(x,y)$. Calculate the local complexity

for all $P_{M_1}^{(0)}$ positions in $img_{P1}$, and sort the prediction error $PE$ according to the local complexity from small to large. If the number of elements in the $P_{M_1}^{(0)}$ set is $N$, that is $|P_{M_1}^{(0)}| = N$, the prediction error sequence $(e_1, e_2, e_3, ..., e_N)$ sorted by local complexity is obtained.

$$\rho(x,y) = \frac{\sum_{(i,j)\in Neighbour(x,y)}(img(i,j) - \overline{Neighbour(x,y)})^2}{|Neighbour(x,y)|} \tag{11}$$

Use parameters $T_l$, $T_r$, $T_d$ to represent the threshold parameters for prediction error expansion, where $T_l < T_r$, $T_d > 0$. If the watermark bit to be embedded is $b \in \{0,1\}$, modify $i$-th prediction error $e_i$ sorted by local complexity $e_i'$ to according to Formula (12).

$$e_i' = \begin{cases} e_i - (T_l - e_i) - b, & \text{if } T_l - T_d < e_i \le T_l \\ e_i + (T_l - e_i) + b, & \text{if } T_r \le e_i < T_r + T_d \\ e_i + T_d, & \text{if } e_i \ge T_r + T_d \\ e_i - T_d, & \text{if } e_i \le T_l - T_d \\ e_i, & \text{otherwise} \end{cases} \tag{12}$$

Finally, pixels in all positions are modified according to the new prediction error to complete the embedding of the watermark and obtain the watermark image $img_{w1}$.

$$img_{w1}(x,y) = \begin{cases} img_{p1}(x,y) + PE'(x,y), & \text{if } \quad (x,y) \in P_{M_1}^{(0)} \\ img(x,y), & \text{if } (x,y) \in P_{M_1}^{(1)} \end{cases} \tag{13}$$

If the watermark information is too large to be fully embedded, a second level embedding is required. When the second level embedding is performed, the watermark is embedded in the same way by multiplying $img_{w1}$ with matrix $M_2$. Finally, the parameters are also stored in the $LSB$ of the zero row.

During the embedding process of watermark information, it is necessary to modify the prediction error $e_i$. After embedding the watermark, it should meet the requirement $0 \le x_i' \le 255$. Since the prediction error can move up to $T_d$, it is necessary to record the pixels $0 \le x < T_d$ and $255 - T_d < x \le 255$ using *Bitmap*. After compressing the *Bitmap*, it is embedded as part of the watermark information into the image to address overflow issues.

Let's take $T_d = 1$ as an example. When $x_i = 0$ and $\widehat{x_i} = 2$, $e_i = -2$ shifts to $e_i = -3$, resulting in $x' = -1$, which generates underflow. Similarly, when $x_i = 255$, there will be an overflow, so *Bitmap* is used to record positions 0 and 255 in all $P_{M_1}^{(0)}$ positions in *img*. Modify the pixel values at positions 0 to 1, 254 to 255, and then embed *Bitmap* compressed as part of the watermark information into the image. At the same time, embed the compressed length of *Bitmap* into the zero line through $LSB$ replacement.

3.4. **Watermark Extracting.** The process of watermark extraction is the inverse of the watermark embedding process, as shown in Figures 8.

Firstly, the current number of watermark layers and parameters $T_l$, $T_r$, $T_d$ as well as the length of the watermark and the size of *Bitmap* need to be extracted from the $LSB$ of the zero line of the image. After wavelet packet transform, if the current number of embedding layers is an odd layer watermark, image will also be dot multiplied with $M_1$ (or $M_2$ if it is an even layer watermark), and input into the model for prediction. After modification, the same predicted image as the watermark embedding can be obtained. The prediction error and local complexity are calculated according to the method used during watermark embedding, and then the prediction error is sorted by local complexity to perform watermark extraction. When extracting watermarks, the original prediction error $e_i$ can be calculated from the modified prediction error $e_i'$ using Formula (14). After

obtaining the original prediction error, the watermark information embedded in the position can be extracted using Formula (15), and the original pixel value $x_i$ can be restored using Formula (16). If the current number of layers is greater than 1, it indicates that multi-level embedding has been performed, repeating the process of watermark extraction until the original image *img* is restored.

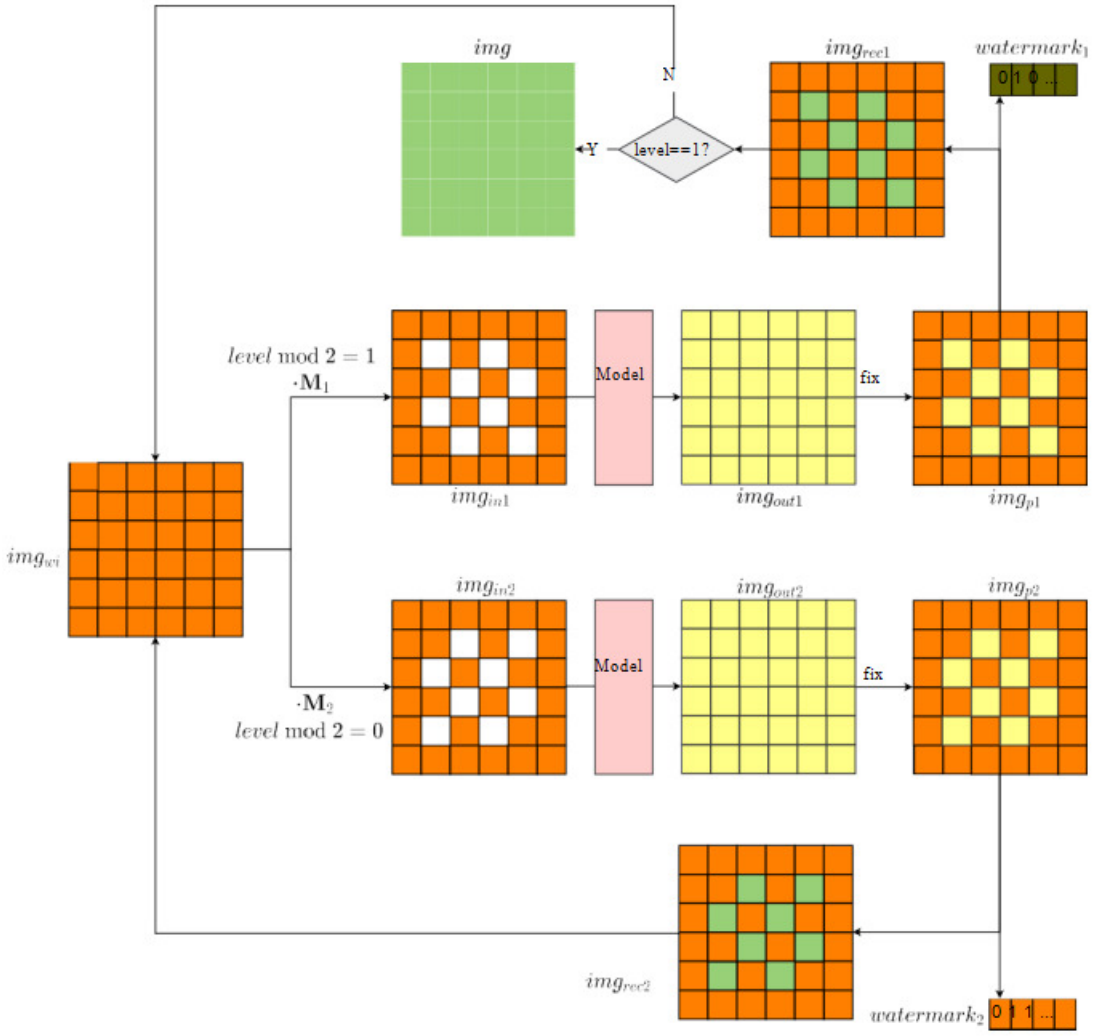



FIGURE 8. Schematic Diagram of Watermark Extraction Steps

$$e_i = \begin{cases} \left\lceil \frac{e'_i+T}{2} \right\rceil, & \text{if } T_l - 2T_d < e'_i \le T_l \\ \left\lfloor \frac{e'_i+T_r}{2} \right\rfloor, & \text{if } T_r \le e'_i < T_r + 2T_d \\ e'_i - T_d, & \text{if } e'_i \ge T_r + 2T_d \\ e'_i + T_d, & \text{if } e'_i \le T_l - 2T_d \\ e'_i, & \text{otherwise} \end{cases} \tag{14}$$

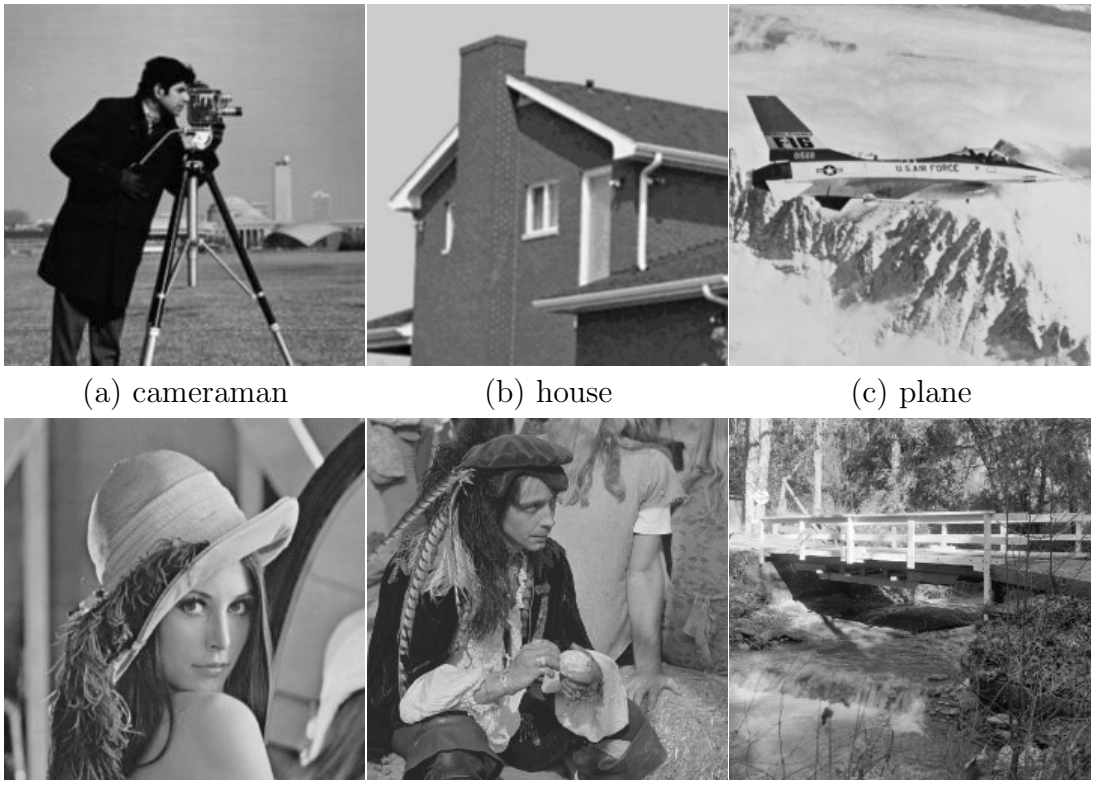
$$b = \begin{cases} 2 \times e_i - T_l - e'_i, & \text{if } T_l - 2T_d < e'_i \le T_l \\ e'_i - 2 \times e_i + T_r, & \text{if } T_r \le e'_i < T_r + 2T_d \end{cases} \tag{15}$$

$$x_i = e_i + \widehat{x_i} \tag{16}$$

After watermark extraction and original pixel restoration, *Bitmap* is extracted from the watermark information based on the *Bitmap* size in zero line $LSB$, and overflow bits are

recovered. At the same time, the zero line original $LSB$ is extracted from the watermark information. If the embedding level of the current watermark $level > 1$, it indicates multi-level embedding. Repeat the above steps for the recovered image until the embedding $level = 1$, indicating that it is watermark information of the last layer. After restoration, original image $img$ is obtained, and the watermark information at each level is combined to obtain the final watermark information $watermark$.

4. **Experiment.** Our proposed scheme was compared with other traditional predictors, they are the methods of Sachnev et al. [11], Jia et al. [13] and Tang et al. [14]. Standard Test Images are used here, as shown in Figure 9.



(a) cameraman (b) house (c) plane

(d) lena (e) pirate (f) walkbridge

FIGURE 9. Standard Test Images

The design method of prediction accuracy experiment is to use different prediction methods based on diamond predictor scheme to predict the test pictures after dividing the test pictures by chessboard. The prediction accuracy is measured by comparing $MSE$ of the predicted pictures and the original pictures. $MSE$ is calculated by Formula (17). The smaller the $MSE$, the higher the prediction accuracy. By comparing the maximum embedding capacity $Capacity_{max}$ to measure the maximum capacity of an image, the maximum embedding capacity defined in this experiment refers to the maximum number of bits embedded when using threshold $T_l = 0$, $T_r = 1$, $T_d = 1$ for watermark embedding in a single-layer prediction scenario, which is the sum of the total number of predicted pixel points with a prediction error of 0, 1, as shown in Formula (18).

$$MSE = \frac{1}{n} \sum_{i,j} (img(i,j) - img_w(i,j))^2 \tag{17}$$

$$Capacity_{max} = Num_{PE=0} + Num_{PE=1} \tag{18}$$

TABLE 1. MSE of Each Method

| | *cameraman* | *house* | *plane* | *lena* | *pirate* | *walkbridge* |
|---|---|---|---|---|---|---|
| Ours | **4.281** | **0.233** | **3.156** | **9.838** | **17.588** | **44.795** |
| Sachnev et al. [11] | 5.486 | 0.909 | 10.665 | 12.157 | 23.756 | 59.729 |
| Jia et al. [13] | 4.785 | 0.761 | 9.469 | 11.524 | 23.252 | 57.145 |
| Tang et al. [14] | 6.321 | 0.758 | 10.412 | 11.842 | 23.123 | 68.930 |

TABLE 2. Average MSE and Capacitymax Results of Each Method on the BOSS Dataset

| | *Average MSE* | *Average* |
|---|---|---|
| Ours | **15.985** | **70012.4** |
| Sachnev et al. [11] | 24.513 | 54906.5 |
| Jia et al. [13] | 22.754 | 59420.9 |
| Tang et al. [14] | 25.378 | 62088.9 |

As shown in Table 1, our proposed method achieved a lower $MSE$ value, the proposed deep learning-based predictor is better than traditional rhombus-based predictor.

This experiment compared the prediction results of several different predictors on six Standard Test Images, and used the line chart to draw the distribution map of prediction error, as shown in Figure 9. It can be seen that the line chart of prediction error generated by the scheme based on diamond prediction and neural network proposed in this chapter is significantly more concentrated and has better prediction effect.

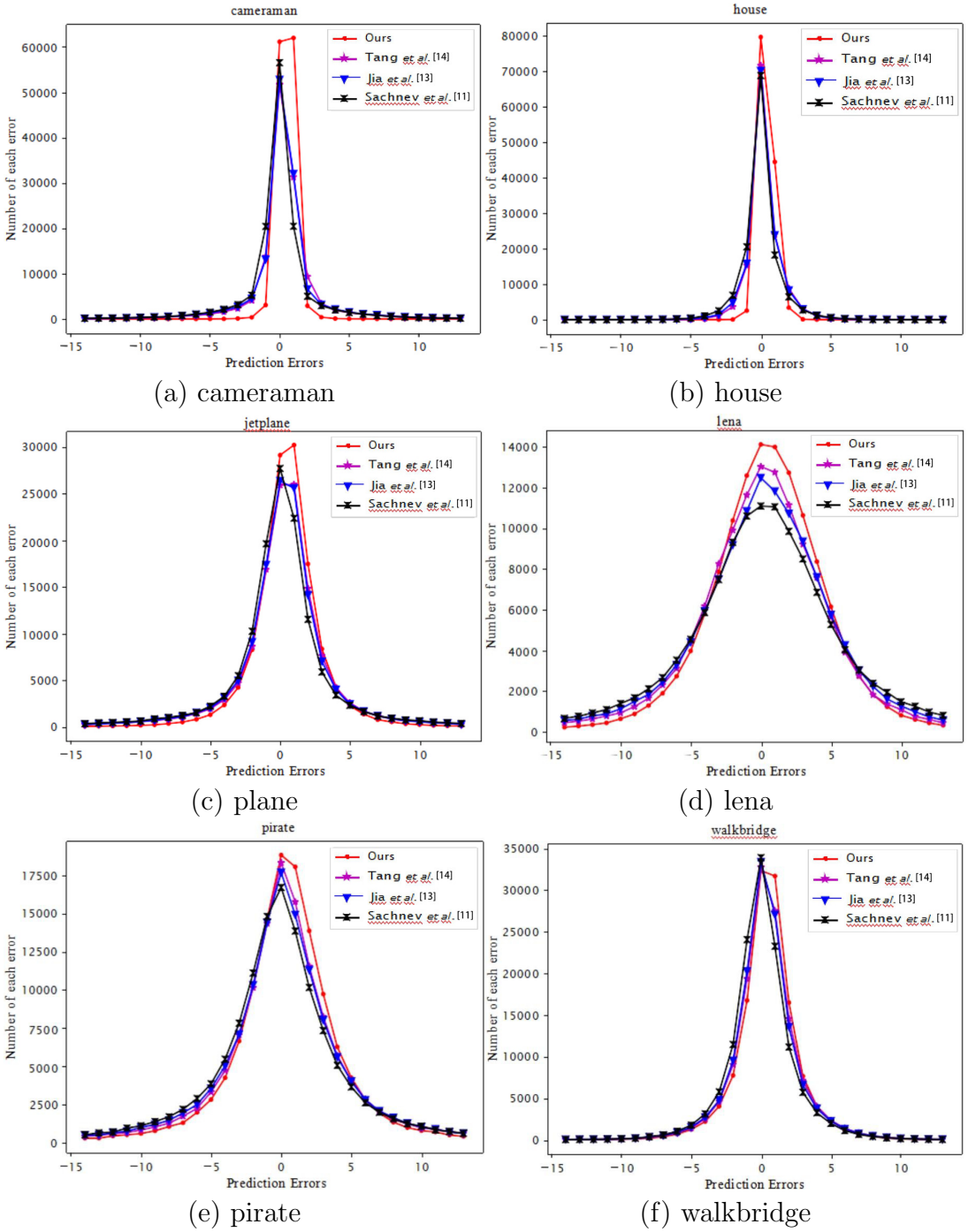
Then, an experiment compares the prediction results of several different predictors on six Standard Test Images are given, and we use line chart to draw distribution map of prediction error, as shown in Figure 10. It can be seen that the line chart of prediction error generated by the scheme based on diamond prediction and neural network proposed in this chapter is significantly more concentrated and has better prediction effect.

In addition, further testing was conducted on 10000 images in the BOSS dataset, and the average values of $MSE$ and $Capacity_{max}$ were taken from the 10000 images. The results are shown in Table 2. Through this comparison, superiority of the predictor proposed is further demonstrated.

The design of the watermark embedding experiment is to use the prediction error results of different prediction schemes compared in the last experiment to perform watermark embedding experiments on standard images from the six Standard Test Images. In order to control variables and demonstrate effectiveness of this method, we use the same embedding threshold parameters for watermarking embedding, that is, a calculated local complexity according to the same local complexity calculation method, and then sorts the local complexity for prediction error expansion and watermark information embedding. The watermark embedding capacity is from $2000bit$ to the maximum single-layer embedding capacity of the image, and the peak signal-to-noise ratio (PSNR) of the watermark image and the original image under different embedding capacities is recorded. The calculation method of PSNR value is shown in Formula (19):

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \tag{19}$$

Here, where $a$ is the maximum possible pixel value of the image, and the color depth of the image used in this experiment is 8 bits, so this value is 255. Finally, the watermark embedding performance of each method under different embedding capacities is shown in Figure 11.



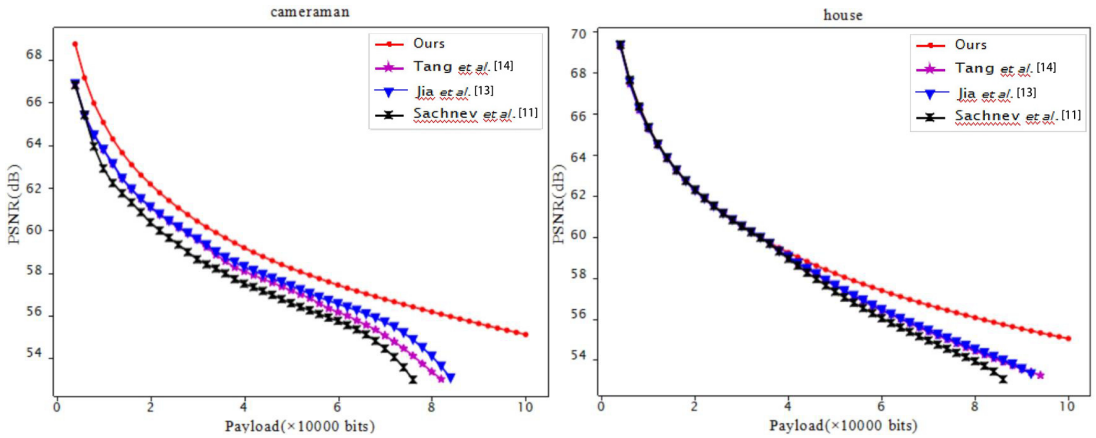

(a) cameraman (b) house

(c) plane (d) lena

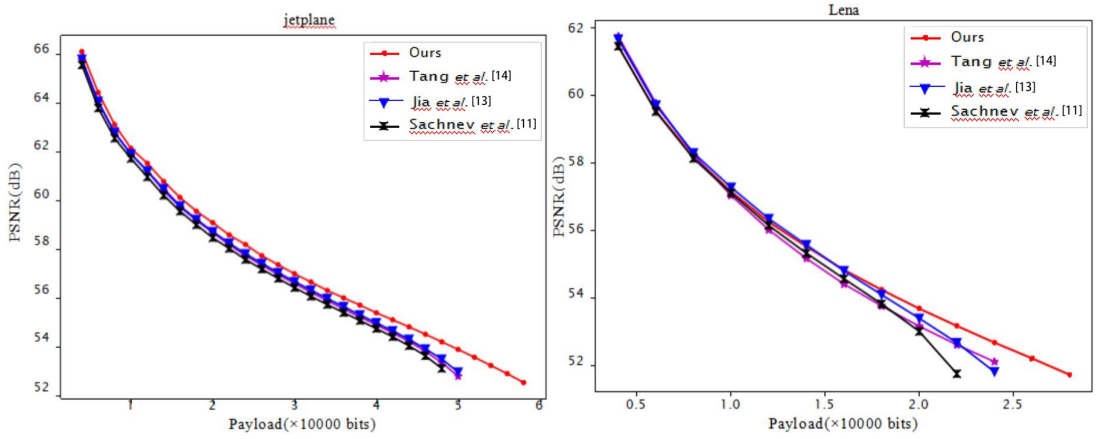
(e) pirate (f) walkbridge

FIGURE 10. Comparison Chart of Prediction Error Distribution

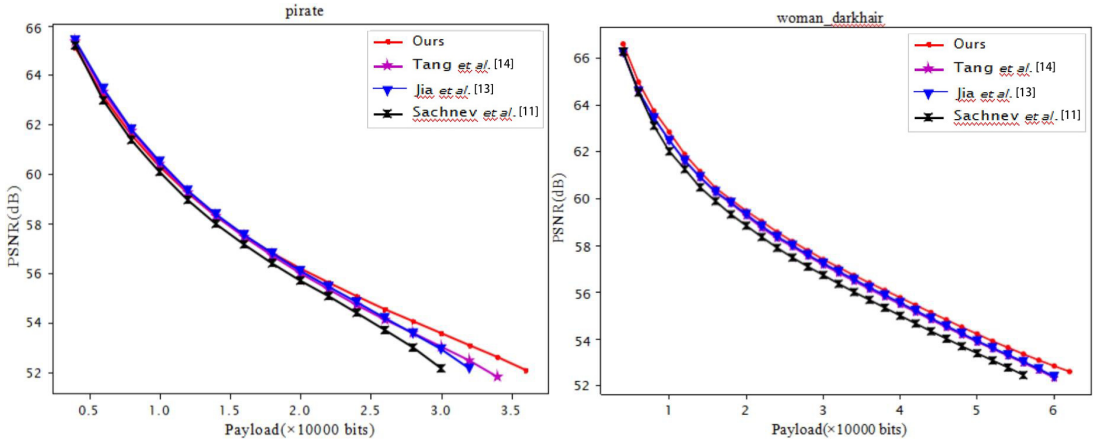
From the experimental results, it can be seen that the maximum embedding capacity of the proposed scheme in this paper is superior to several traditional predictors, regardless of whether it is a simple image or a complex image. In terms of embedding performance, scheme proposed has a significant effect on simple images (cameraman). Whether it is low or high embedding capacity, watermark embedding performance is superior to several traditional predictors, especially in high embedding capacity. On complex images, the scheme proposed in this paper performs similarly to traditional methods at low embedding capacity, while it performs very well at high embedding capacity. Under the same




(a) Cameraman (b) House




(c) Plane (d) Lena




(e) Pirate (f) Walkbridge

FIGURE 11. Watermark Embedding Performance Comparison

distortion premise (with the same PSNR value), the scheme proposed in this paper can significantly embed a larger capacity.

5. **Conclusions.** A new predictor based on deep neural network combines deep learning and Diamond Predictor has been proposed. It has achieved high accuracy. After wavelet packet decomposition, watermark embedding process has been applied on the coefficients at different levels. ImageNet dataset has been used as the test set. Experiment shows that the proposed predictor is good, it achieves higher embedding capacity and less distortion.

**Acknowledgment.** This work is supported by school research of Laiwu Vocational and Technical College named Application Research of Watermarking Frame Algorithm and Wavelet Packet Decomposition Based on Genetic Algorithm. And it is also supported by Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005). This research is also supported by Key Basic Research Projects of Shenzhen with Grant No. JCYJ20220818102414030.

## REFERENCES

[1] J. Fridrich, M. Goljan, R. Du, "Invertible authentication," in *Security and Watermarking of Multimedia contents III*. SPIE, 2001, vol. 4314, pp. 197-208.

[2] Z. Ni, Y.-Q. Shi, N. Ansari, W. Su, "Reversible data hiding," *IEEE Transactions on Circuitous and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362, 2006.

[3] Y. Ma, Y. Peng, T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121-2133, 2022.

[4] F.-Q. Zhang, T.-Y. Wu, G.-Y. Zheng, "Video salient region detection model based on wavelet transform and feature comparison," *EURASIP Journal on Image and Video Processing*, vol. 2019, 58, 2019.

[5] F.-Q. Zhang, T.-Y. Wu, Y.-O. Wang, R. Xiong, G.-Y. Ding, P. Mei, L.-Y. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[6] E.-K. Wang, C.-M. Chen, M.-M. Hassan, A. Almogren, "A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain," *Future Generation Computer Systems*, vol. 108, pp. 135-144, 2020.

[7] K.-K. Tseng, J.-R. Lin, C.-M. Chen, M.-M. Hassan, "A fast instance segmentation with one-stage multi-task deep neural network for autonomous driving," *Computers & Electrical Engineering*, vol. 93, 107194, 2021.

[8] D.-M. Thodi, J.-J Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721-730, 2007.

[9] D.-M. Thodi, J.-J Rodríguez, "Prediction-error based reversible watermarking," in *2004 International Conference on Image Processing*. ICIP, 2004, vol. 3, pp. 1549-1552.

[10] X. Qu, S. Kim, H.-J. Kim, "Reversible watermarking based on compensation," *Journal of Electrical Engineering and Technology*, vol. 10, no. 1, pp. 422-428, 2015.

[11] V. Sachnev, H.-J. Kim, J. Nam, S. Suresh, Y.-Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999, 2009.

[12] H. Zhao, O. Gallo, I. Frosio, J. Kautz, "Loss functions for neural networks for image processing," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47-57, 2017.

[13] Y. Jia, Z. Yin, X. Zhang, Y.-L. Luo, "Reversible data hiding based on reducing invalid shifting of pixels in histogram shifting," *Signal Processing*, vol. 163, pp. 238-246, 2019.

[14] X. Tang, L. Zhou, D. Liu, B.-Y. Liu, "Reversible data hiding based on improved rhombus predictor and prediction error expansion," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*. TrustCom, 2020, pp. 13-21.