Journal of Computer and Communications, 2026, 14(11), 184-203
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Study on Fragile Watermarking-Based Tamper-Proof Image Authentication for Industrial X-Ray Image

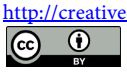
Peichen Zheng[1], Shugang Cheng[1], Bing He[2,3*], Shijie Wang[2], Jing Wang[3], Gengwu Chen[3]

[1]Chongqing Unicomp Technology Co., Ltd, Chongqing, China
[2]School of Physics and Electrical Engineering, Weinan Normal University, Weinan, China
[3]Engineering Research Center of X-Ray Imaging and Detection, Universities of Shaanxi Province, Weinan, China
Email: *hebing@wnu.edu.cn

How to cite this paper: Zheng, P.C., Cheng, S.G., He, B., Wang, S.J., wang, J. and Chen, G.W. (2026) A Study on Fragile Watermarking-Based Tamper-Proof Image Authentication for Industrial X-Ray Image. *Journal of Computer and Communications*, **14**, 184-203.
https://doi.org/10.4236/jcc.2026.1411009

Received: November 2, 2026
Accepted: November 17, 2026
Published: November 20, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/



Open Access

Abstract

Industrial X-ray images must simultaneously guarantee the integrity of the original pixel information and the detectability of local content tampering during storage and transmission. Existing block-mean-based authentication methods effectively mitigate the cross-block false alarm problem caused by image-level normalization through a fixed-range mapping mechanism. However, the authentication features of such methods can only reflect the mean information of image blocks, and thus suffer from a detection failure limitation for image detail tampering that preserves the block mean. To address this deficiency, this paper proposes a block-level authentication method that integrates a hash-based message authentication code (HMAC) with carrier consistency verification. The proposed method incorporates all pixel values within a 4 × 4 RGB image block, the image identifier, and the block position information into the authentication code generation process, and embeds the 24-bit tag in the first decimal place of selected discrete Hartley transform (DHT) coefficient magnitudes. At the receiver side, the authentication codes of candidate image blocks are verified, and the consistency between the received carrier and the reconstructed carrier is simultaneously checked. On a single industrial casting X-ray image, the block-mean baseline's block-level missed detection rates differed between the two tested attacks, with 0.22% under 5% salt-and-pepper noise and 0.00% under contiguous white-region tampering. On the same image, HMAC detected a mean-preserving edit and the carrier check detected a small float64 perturbation. The joint check localized the target block in both cases.

Keywords

Fragile Watermarking, Image Authentication, Carrier Consistency,

Tamper-Proof Verification

1. Introduction

Industrial X-ray images primarily record defect morphology, location, and grayscale variations used for image interpretation, so their integrity directly affects the reliability of inspection conclusions. Images may be illegally tampered with during storage, transmission, or archiving; therefore, content changes must be identified while preserving pixel information. Industrial X-ray images reflect differences in internal materials and structures of workpieces through X-ray attenuation, and their grayscale formation mechanism differs from that of ordinary visible-light natural images. Unlike medical images that target human tissues and anatomical structures, this study focuses on the internal structures of industrial castings and their local grayscale and edge details. Regions containing the same material or regular components may exhibit similar grayscale values. Defects such as pores must be interpreted through local grayscale and edge details, and tampering with these details may affect inspection conclusions. A change in one detail does not necessarily alter the average grayscale of the image block in which it is located; therefore, authentication based solely on block means may miss such tampering. Related studies have developed reversible information hiding [1], difference expansion [2], reversible authentication [3], block-wise authentication [4], prediction error methods [5], and block feature authentication [6]. Fragile watermarking can sense changes in watermarked images, while reversible mechanisms can restore the host image after extracting authentication data. The combination of the two can address the issues of image integrity and pixel-level recovery.

Most studies concentrate on finding the altered region and on whether that localization is stable. Hierarchical authentication refines the reported location [7]. Chaotic keystreams, block features, and self-embedding are used to strengthen the security and the discriminability of the authentication data [6] [8]. Other schemes embed the data in a transform domain so that the watermark responds more directly to content changes [9]. As the same ideas have been carried over to medical images, fragile authentication, tamper localization, and host recovery have been placed in one pipeline, so that integrity checking and restoration proceed together [10] [11]. Dual detection has also been introduced: different decisions are cross-checked in order to improve reliability [12]. Transform-domain features, dependence between blocks, and reversible embedding now form a fairly clear technical path. False alarms and missed blocks can still depend jointly on the gray-level distribution, the block partition, and the decision rule.

Recent work gives more weight to reversibility, fine localization, and designs that serve more than one purpose. Bouarroudj *et al.* encrypt DCT authentication data with a Fibonacci Q-matrix and embed them in DFT coefficients, obtaining high-fidelity reversible authentication of medical images [13]. Yan *et al.* combine

a fragile watermark with a zero watermark for region-of-interest (ROI) integrity verification, tamper localization, and copyright protection [14]. Jagadeesh *et al.* use the slant transform, DNA coding, and chaotic block mapping for multi-region localization and self-recovery [15]. Later schemes extend authentication and recovery through multi-feature codes with cross-level embedding [16], hybrid transforms with dual self-recovery [17], and ROI-partitioned reversible embedding [18]. These designs focus on feature construction, encryption, and the mapping of recovery data. Less attention is given to cross-block false alarms caused by image-level normalization, and to small regions removed by neighborhood filtering.

These studies provide different approaches to authentication and recovery. Fixed-range mapping avoids cross-block dependence introduced by image-level normalization, but its authentication feature is still determined by the block mean and cannot distinguish blocks with the same mean and different pixel distributions. To address this limitation, this paper proposes block-level authentication combining complete-block HMAC with a carrier-consistency check, which separately verifies all pixel contents of a candidate block and consistency between the received and reconstructed carriers. The work reported here concerns the applicability and limitations of block-mean authentication: 1) the roles of fixed-range mapping and isolated anomalous-block retention are analyzed, explaining how they avoid cross-block dependence from image-level normalization and suppression of isolated anomalies by neighborhood-majority filtering, while identifying the limitation of a block-mean feature; 2) a single industrial X-ray image is used for no-tamper, 5% salt-and-pepper noise, and contiguous white-region tampering experiments, with embedding fidelity, authentication-data extraction, and block-level localization results reported to examine responses to different forms of tampering. We also evaluate the two checks on the same test image using a mean-preserving edit and a small carrier perturbation. The computational and storage costs of different authentication approaches require further experimental validation.

2. Local Frequency-Domain Authentication and Analysis

2.1. Baseline Framework

The baseline follows a DWT-Logistic-DHT pipeline. Every input is converted to RGB. The original 3072 × 3072 image is downsampled to 1024 × 1024 using Lanczos resampling, without cropping or geometric distortion. The image is partitioned into non-overlapping 4 × 4 blocks, which are the smallest units of tamper localization. Each color channel then undergoes a two-level orthonormal Haar decomposition, and the low-frequency sub-band LL_2 is retained. Quantization of the LL_2 coefficient produces one 8-bit authentication byte for each channel block. That byte is combined, by bitwise exclusive-or, with a Logistic byte stream initialized from the key K and the channel index. Compute SHA-256 of the UTF-8 string K + NUL + str(c), where NUL denotes one 0x00 byte and $c = 0,1$, or 2 denotes R, G, or B, respec-

tively. Read the first eight digest bytes and the next eight as unsigned big-endian integers u_0 and u_1. Set $x_0 = (u_0 + 0.5)/2^{64}$ and $r = 3.9 + 0.099999 \times u_1/2^{64}$. Iterate $x_{n+1} = r x_n (1 - x_n)$ in float64 and discard the first 1000 iterations. Each subsequent iteration produces $\text{floor}(256 x_n) \bmod 256$. Fill each channel in block row-major order. The Logistic sequence is used here only to process authentication data. It is not offered as a standard cipher, and the security of the system still depends on key management, parameter setting, and the implementation environment.

At embedding time, each 4 × 4 block is transformed by a two-dimensional discrete Hartley transform, and the low-frequency component is shifted to the center of the transform domain. The unnormalized DHT is the real part minus the imaginary part of the two-dimensional FFT. After shifting both axes, the DC coefficient is at (2, 2). The authentication byte is written into eight preselected boundary coefficients, most significant bit first. The zero-based row and column positions are (0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 3), (2, 0), and (2, 3), in that order. The embedded coefficient is

$$c' = s\left[\lfloor |c| \rfloor + 0.1b + 0.1d\right] \tag{1}$$

where c and c' are the real transform coefficients before and after embedding, s is the sign of the coefficient (positive for zero), $b \in \{0,1\}$. is the authentication bit, and d is the fractional part of |c|. The receiver reads the leading decimal digit of each selected coefficient, maps that digit back to a bit, and shifts the remaining fractional digits into their original places so that the pre-embedding coefficient is restored. A leading digit of 1 is decoded as bit 1, and any other digit is decoded as bit 0. In the implementation, the absolute value of each selected coefficient is first rounded to nine decimal places. For a rounded magnitude with fractional part f, the leading digit is floor (10 f + 10^{-9}). Digits outside the expected set are only counted; the block decision does not depend on them. The inverse transform undoes the shift, applies the same DHT, and divides by 16.

Pack the extracted bits in embedding order and XOR with the same Logistic stream to recover qext. Before coefficient restoration, round the received spatial-domain samples with ties to even, clip them to [0, 255], and convert them to uint8. Use these samples to recompute qrecv = round (LL_2/4), again with ties to even. The extracted authentication byte is then compared, block by block, with the byte regenerated from the received image. Their maximum absolute difference across the three channels is the block anomaly $D(i, j)$. The block is declared anomalous when $D(i, j) > T$ and normal otherwise. All block-mean baseline experiments use $T = 5$. The threshold is the tolerated difference between feature bytes, and it is kept fixed for all reported baseline conditions.

2.2. Fixed-Range Feature Mapping

For comparison, image-wise min-max mapping normalizes each low-frequency feature by the minimum m and the maximum M observed on the current image,

and then quantizes the LL_2 coefficient:

$$q_{mm} = \text{round}\left(\frac{255(a-m)}{M-m}\right) \tag{2}$$

where a is the low-frequency coefficient to be quantized, and m and M are the minimum and maximum observed on the current image. Because both extremes depend on the whole image, a local edit can change the global quantization scale. An untouched coefficient a may then be quantized to a different byte, and the mismatch is reported as a cross-block false alarm.

For an arbitrary 4 × 4 host block B, the low-frequency coefficient after the two-level orthonormal Haar decomposition is

$$LL_2 = \frac{1}{4}\sum_{(u,v)\in B} p_{uv}, \quad 0 \le LL_2 \le 1020 \tag{3}$$

where p_{uv} is the pixel at (u, v) in B. With $p_{uv} \in [0, 255]$, and the Haar scaling used here, LL_2 lies in [0, 1020]. The bounds follow from the transform and from the range of an 8-bit sample. Quantization on this fixed interval is

$$q_{\text{fix}} = \text{round}\left(\frac{LL_2}{4}\right), \quad 0 \le q_{\text{fix}} \le 255 \tag{4}$$

Together with Equation (3), the fixed-range feature equals the rounded mean gray level of the 4 × 4 block, and q_{fix} is the resulting 8-bit authentication byte. The mapping removes the cross-block dependence of image-level min-max normalization, and it does not require the extremes to be transmitted or re-estimated. A local edit then appears in the corresponding LL_2 coefficient and in its authentication byte, rather than shifting the scale used by other blocks. The feature is still only a block mean, so a local change that preserves the mean is not identified.

Fixed-range mapping is intended for integrity checking on one pixel grid. It does not claim robustness to JPEG compression, scaling, or rotation. Those operations can change many blocks at once and can destroy the original block alignment. For a fragile authenticator they are content changes to be detected and reported, not distortions to be tolerated as in robust watermarking.

2.3. Retention of Isolated Anomalous Blocks and Decision Measures

After the block anomaly map is obtained, the baseline applies a neighborhood-majority rule. The revised procedure drops that post-processing and keeps every block that the threshold has marked. Against a ground-truth tamper mask, true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) are counted, and the false-positive rate (FPR), false-negative rate (FNR), and F1 score are computed from them.

The false-positive rate is the fraction of untouched blocks declared tampered:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

The false-negative rate is the fraction of truly tampered blocks declared normal:

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FN} + \mathrm{TP}} \tag{6}$$

F1 summarizes precision and recall through the equivalent confusion-matrix form:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

FPR and FNR describe, respectively, false alarms on clean blocks and misses on tampered blocks. F1 combines precision and recall, but it is sensitive to class balance when tampered blocks are few, so FPR, FNR, and F1 are reported together. Under no attack, the peak signal-to-noise ratio (PSNR), the bit-error rate (BER), the number of false-alarm blocks, and the recovery error are also recorded, in order to judge fidelity and the stability of the authentication data. Keeping the raw anomalous blocks avoids deleting an isolated block or a thin tampered line. That choice, together with fixed-range mapping, is the configuration used in the experiments below.

2.4. Reversible Recovery and Complexity

The spatial-domain float64 watermarked image array is the authentication carrier. Once the authentication bits have been read, the leading fractional digit that carried each bit is removed and the remaining digits are shifted back, which restores the pre-embedding values of the selected DHT coefficients. In the no-attack experiment the BER of the authentication data is zero, and the software inverts the embedding. Recovery is evaluated on the float64 carrier. The high-precision carrier and the display image are kept distinct: the former is used for extraction and recovery, and the latter is obtained by rounding and clipping the former to 8 bits for display only. Quantization discards the fractional digits that carry the authentication bits, so a display image cannot stand in for the carrier. Recovery is measured against the host after conversion to RGB and any size adjustment, and any file that is saved or reloaded must preserve the float64 array.

For an image of height H and width W, the two-level Haar transform, authentication-code generation, the discrete Hartley transform, and the 4×4 partition all operate on pixels. Each block has a constant transform size and a constant number of steps, so the cost grows linearly with the number of pixels. The Logistic keystream is proportional to the length of the authentication data and is likewise linear. The time and space complexity are therefore both O(HW). Relative to min-max mapping from observed extremes, fixed-range mapping changes only the quantization of the low-frequency feature. It adds neither an iteration nor a data structure, and it does not change the asymptotic complexity.

2.5. Complete-Block Authentication and Carrier Consistency

Serialize each 4×4 RGB block as IW01 (4 ASCII bytes), the image-version identifier (16 bytes), the zero-based row and column (4 unsigned big-endian bytes

each), and 48 uint8 samples (row-major, interleaved RGB). The receiver obtains the expected identifier from its configuration.

Take the first 24 bits of HMAC-SHA-256 [19] with a separate 32-byte key. Eight DHT positions per channel provide 24 bits per RGB block. Embed one tag byte per channel in RGB order, most significant bit first, using Section 2.1 and Equation (1).

Extract t from the received float64 block Y as in Section 2.1. Restore each selected coefficient to $s\left[\text{floor}(m)+\text{frac}(10m)\right]$, where m is its magnitude rounded to nine decimal places, frac gives the fractional part, and s is its sign (positive for zero). Invert the DHT, round with ties to even, and clip to $[0,255]$ to obtain uint8 candidate $\hat{B}$.

The HMAC check compares t with the tag recomputed from $\hat{B}$. Re-embed t in $\hat{B}$ to reconstruct $\hat{Y}$. The carrier check requires valid leading digits (0 or 1) and $\delta=\max\left|Y-\hat{Y}\right|\le 10^{-8}$ over the 48 float64 samples. Accept the block only if both checks pass.

3. Interface Design and Implementation

A PyQt5 front end organizes host loading, key entry, watermark generation and embedding, extraction from a received image, tamper localization, and attack evaluation. **Figures 1-3** show the PyQt5 interface for the block-mean baseline. The HMAC prototype was evaluated in a standalone implementation.

3.1. Layout

A parameter bar sits above a tabbed workspace. The bar selects the host, sets the key K, and offers generation, saving, embedding, and saving of the watermarked image. The tabs are, in order, watermark generation, embedding, extraction and tamper detection, 5% salt-and-pepper attack, cropping attack, and algorithm steps. **Figure 1** shows generation, **Figure 2** shows 4 × 4 embedding and the difference image, and **Figure 3** shows extraction, localization, and recovery when nothing has been altered.

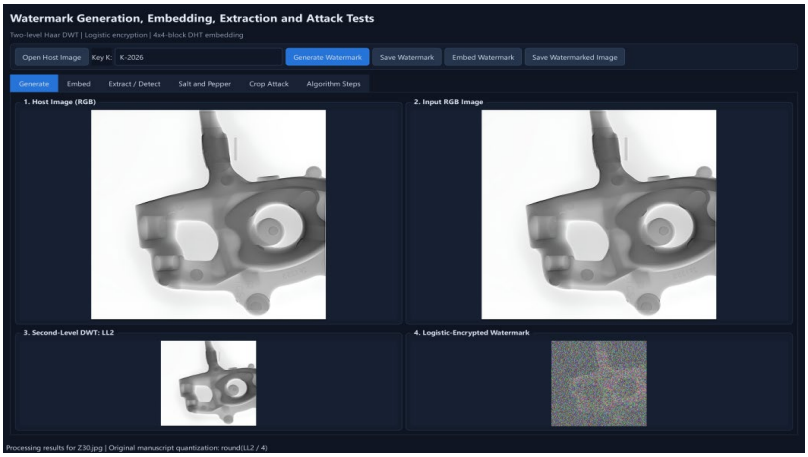


Figure 1. Watermark generation for an industrial casting X-ray image.

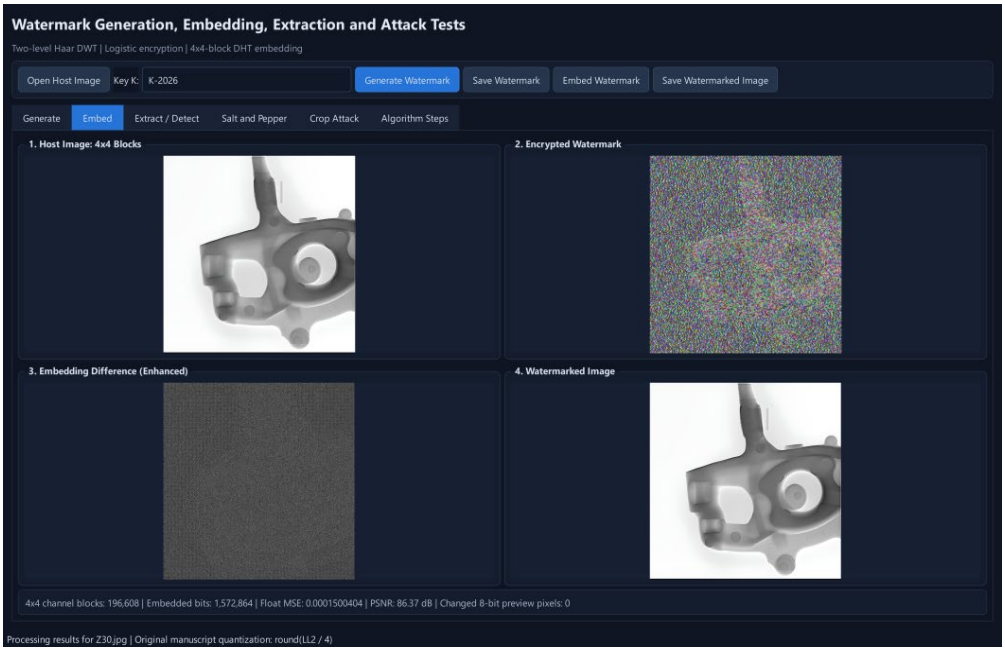


Figure 2. Watermark embedding on 4 × 4 blocks.

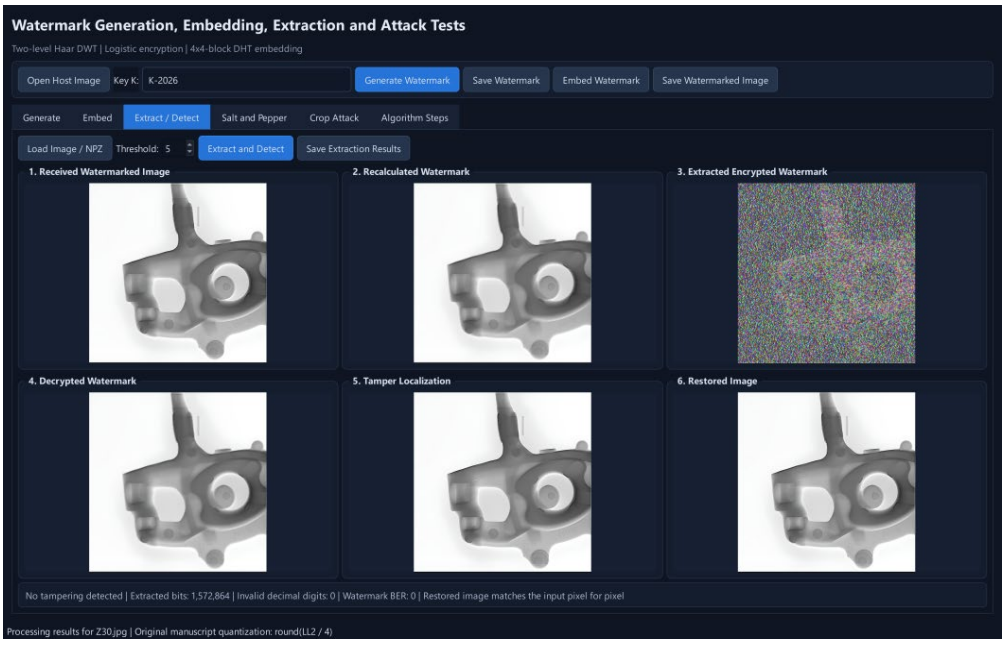


Figure 3. Extraction, localization, and recovery with no tampering.

3.2. Modules and Interaction Constraints

Each module corresponds to one stage of the algorithm, and button states enforce the order of operations. Embedding is enabled only after generation, and extraction and the attack trials are enabled only after embedding. The high-precision carrier is saved as an NPZ file together with an 8-bit preview, so that a quantized display image is not later treated as the reversible carrier. Table 1 lists the modules and their constraints.

Table 1. Software modules and interaction constraints.

Module	Main input	Core output	Interaction and constraint
Generation	Host image, key K	Input RGB image, LL_2, encrypted watermark	Industrial casting X-ray image is selected; embedding and saving open after generation
Embedding	Host, encrypted watermark	Difference image, watermarked image, NPZ	Embedded bit count, MSE, and PSNR are displayed
Extraction and detection	Received image or NPZ, threshold	Extracted watermark, localization map, recovered image	Threshold is editable; execution stays disabled until embedding is finished
Salt-and-pepper attack	Noise ratio, random seed	Attacked image, mask, confusion map	PSNR, BER, NC, and localization measures are written together
Cropping attack	Cropped image, change tolerance	Reconstructed image, localization and class maps	The crop is matched automatically and eight process views are shown

The reproducible settings are key K = K-2026, detection threshold 5, salt-and-pepper ratio 5.0%, random seed 2026, and an internal change tolerance of 3 for the cropping trial. Numeric controls restrict the input range. Buttons are disabled while a stage is running, and the status bar reports the current stage.

3.3. Result Display

Each attack tab shows the image before and after the attack, the pixel-level ground-truth change mask, the block-level localization overlay, the confusion-class map, and the binary localization map. TP, FP, FN, and TN are drawn in green, red, yellow, and black, respectively.

4. Experimental Design

4.1. Images and Protocol

Experiments were performed on Windows 11 using Python 3.12.14, NumPy 2.3.5, and Pillow 12.3.0, with key K-2026 and decision threshold 5. The experimental image was selected from an industrial casting X-ray image collection collected and organized by the authors. The selected sample is the 30th image in the Org_imgs directory (Z30.jpg), with an original size of 3072 × 3072 pixels and an experimental input size of 1024 × 1024 pixels. Figure 4 shows selected images from the industrial casting X-ray image collection; all quantitative results reported here come from the test image.

The experimental sequence comprises watermark generation, 4 × 4 block embedding, extraction without tampering, a 5% salt-and-pepper noise attack, and cropping or region tampering. Watermark generation uses a two-level Haar DWT and Logistic chaotic encryption. During embedding, a DHT is applied to each block and authentication data are embedded in the selected transform coefficients. The watermarked image is saved as both a float 64 high-precision NPZ carrier and an 8-bit PNG preview; the former is used for reversible extraction and the latter for display. The salt-and-pepper noise experiment uses a noise ratio of 5.0% and random seed 2026. In the cropping or region-tampering experiment, a contiguous

white region is introduced using the reconstruction procedure below, and the interface reports an actual changed-pixel proportion of 13.07%. This is a single-image case study, so means, standard deviations, p-values, and effect sizes are not calculated. All baseline quality and localization measures are reported from one deterministic run.

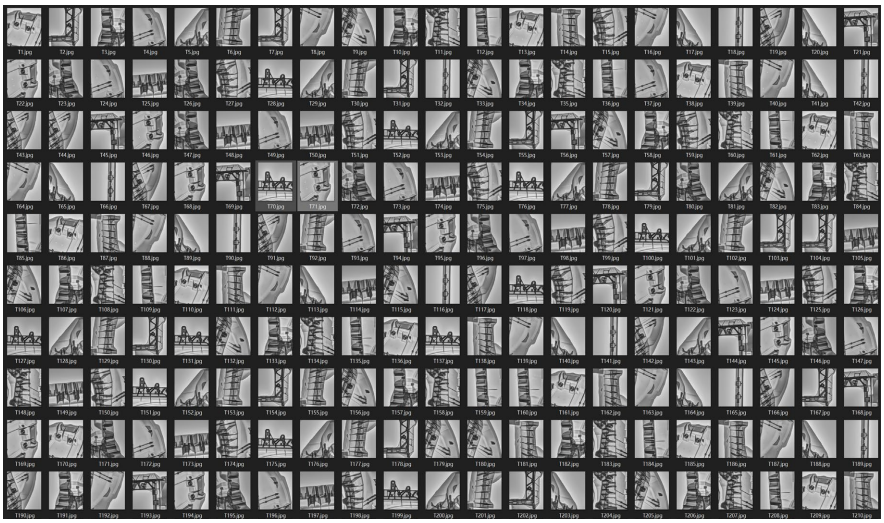


Figure 4. Preview of selected images from the industrial casting X-ray image collection.

Use NumPy default_rng (2026) to sample round (0.05 × 1024 × 1024) = 52,429 pixel locations without replacement. Shuffle 26,214 zeros and 26,215 values of 255 with the same generator, then assign one value to all three channels at each selected pixel. Mark a pixel if any of its float64 samples changed, giving 52,411 changed pixels (5.00%).

In the full-frame 8-bit preview, set rows [288, 633) and columns [512, 912) to 255 in all channels, using zero-based, half-open coordinates. Align the uploaded image with the reference preview at (0, 0). Where the maximum absolute RGB difference exceeds 3, mark the pixel and copy all three uploaded values into the float64 carrier. Retain the original float64 values elsewhere. This yields 137,099 changed pixels. Uploading the unmodified preview reconstructs an identical float64 carrier.

A non-overlapping 4 × 4 block is marked as tampered in the ground truth if any of its pixels is marked. The detector receives the resulting float64 carrier, the 1024 × 1024 dimensions, key K, and threshold 5. The original carrier and ground-truth masks are used for constructing and scoring the attacks. The original feature bytes are used to calculate BER and NC.

Generate separate baseline and HMAC carriers from the same preprocessed test image and verify both unchanged in the control. Both attacks target block (128, 128). Coordinates are zero-based (row, column) pairs. The mean-preserving edit swaps pixels (512, 512) and (514, 515), whose RGB values are (175, 175, 175) and (172, 172, 172). Each channel sum remains 2775. Re-embed each carrier's original

extracted bits in the edited block. For the perturbation, add 10^{-5} to all 16 float64 samples in its red channel.

For reproducibility, use image-version identifier 62958219D97B7B26F10639A C0B194F48, decoded from hexadecimal to 16 bytes. The public test key is the raw 32-byte SHA-256 digest of the ASCII string "Z30 HMAC prototype public test key v1".

The supplementary tests used Python 3.12.14, NumPy 2.3.5, and Pillow 13.0.0. The preprocessed pixel array was identical to the one used in the original baseline experiment.

4.2. Evaluation Measures

Fidelity after embedding, accuracy of the extracted authentication data, and recovery accuracy are measured by the mean squared error (MSE), PSNR, BER, and the maximum absolute recovery error $E_{\max}$ in the floating-point domain.

MSE is the mean squared pixel difference between the original image and the image after embedding:

$$\mathrm{MSE} = \frac{1}{3HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{c=1}^{3}\left[I_w(i,j,c) - I(i,j,c)\right]^2 \tag{8}$$

where I and I_w are the original image and the image after embedding, H and W are the height and width, and i, j, and c index the row, the column, and the color channel. A smaller MSE means a smaller pixel change due to embedding.

PSNR measures the fidelity of the embedded image:

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{9}$$

where 255 is the peak value of an 8-bit sample and MSE is given by Equation (8). PSNR is reported in decibels. A larger value means that embedding changes the visual appearance less.

Embedding PSNR compares the preprocessed host with the embedded carrier. Attack PSNR compares the pre-attack carrier with the received carrier supplied to the detector. Recovery PSNR compares the recovered float64 image with the preprocessed host cast to float64. All three measures use spatial-domain RGB samples in float64 and a peak value of 255.

BER is the fraction of authentication bits that disagree between generation and extraction:

$$\mathrm{BER} = \frac{N_{\mathrm{error}}}{N_{\mathrm{total}}} \tag{10}$$

where N_{error} is the number of incorrect bits and N_{total} is the length of the authentication data. A smaller BER means more accurate extraction.

The maximum absolute recovery error compares the recovered image with the original on the high-precision carrier:

$$E_{\max} = \max_{i,j,c}\left|I_r(i,j,c) - I(i,j,c)\right| \tag{11}$$

where I_r is the recovered image. The maximum is taken over the absolute differences between the recovered float64 samples and the preprocessed host samples cast to float64. A smaller $E_{\max}$ means a closer recovery. Because clean blocks usually dominate the image, an overall accuracy is inflated by the large TN count and does not by itself describe local localization. FPR, FNR, and F1 from Equations (5)-(7) therefore remain the primary localization measures, and they are read together with MSE, PSNR, BER, and $E_{\max}$.

All baseline measures in this round are obtained from one deterministic run on the test image. FPR, FNR, and F1 are computed from TP, FP, TN, and FN. Under no tampering there is no positive block, so precision, recall, and F1 are not interpreted as detection performance. The changed-pixel count, the block-level confusion counts, and the rate measures are reported together, rather than as a single score.

5. Experimental Results and Analysis

5.1. Embedding Quality and Reversibility

The 4 × 4 partition of the test image produces 196,608 channel blocks and embeds 1,572,864 authentication bits. The interface reports a floating-point embedding MSE of 0.0001500404 and an embedding PSNR of 86.37 dB. After quantization, the 8-bit preview has no changed pixel. Extraction with no tampering gives BER = 0. In the untampered control, the recovery PSNR was 339.99 dB and the maximum absolute recovery error was 5.6843×10^{-14}. After rounding to 8-bit values, the recovered image matched the preprocessed host at every pixel and channel. Both the raw watermark similarity and the parameterized similarity are 100%, and the numbers of invalid leading decimal digits and localized watermark pixels are zero.

5.2. Five-Percent Salt-and-Pepper Noise

The 5% salt-and-pepper trial changes 52,411 pixels, which is 5.00% of the image. The attack PSNR is 17.05 dB, the watermark BER is 0.2126, and normalized correlation (NC) is 0.9333. Block localization yields TP = 36,433, FP = 0, TN = 29,023, and FN = 80, corresponding to FPR = 0.00%, FNR = 0.22%, precision = 100.00%, recall = 99.78%, and F1 = 0.9989. Table 2 and Table 3 give the complete measures, and Figure 5 shows the attack, ground-truth mask, localization classification, and binary localization results.

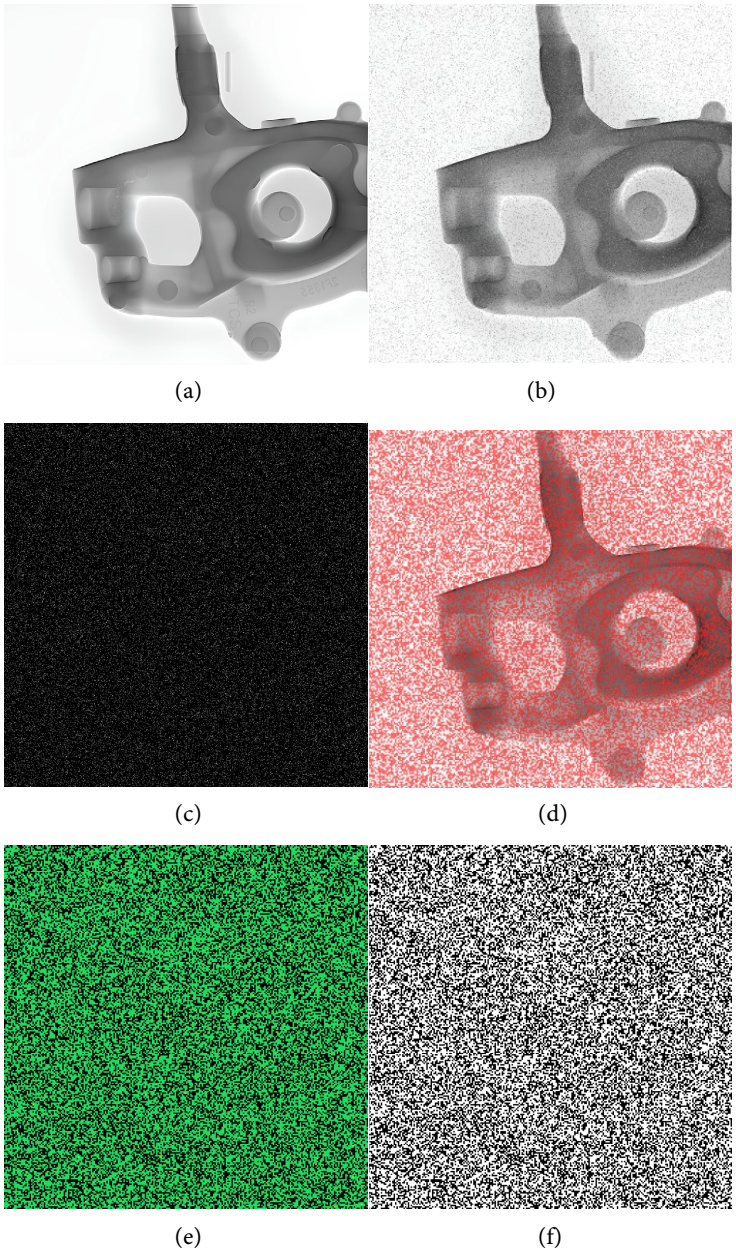
Table 2. Attack conditions and image or watermark quality.

Condition	True change	Attack PSNR (dB)	BER	NC
No tampering	0	∞	0	1.0000
5% salt-and-pepper noise	5.00%	17.05	0.2126	0.9333
White-region replacement	13.07%	15.70	0.0653	0.9865

Table 3. Block-level tamper localization.

Condition	TP	FP	TN	FN	FPR	FNR	Precision	Recall	F1
No tampering	0	0	65,536	0	0	—	—	—	—
5% salt-and-pepper noise	36,433	0	29,023	80	0	0.22%	100%	99.78%	0.9989
White-region replacement	8697	0	56,839	0	0	0	100%	100%	1.0000

Note. Dashes indicate undefined metrics in the untampered control, which had no positive ground-truth blocks or positive detections.

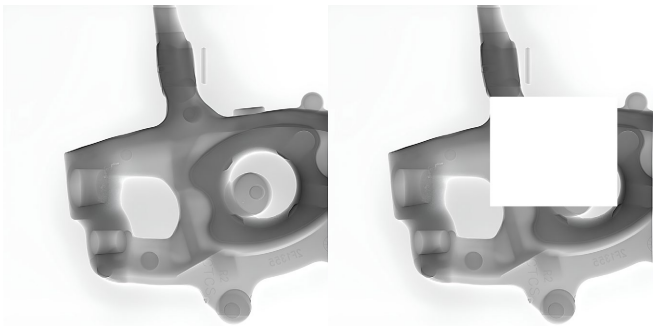


(a) (b) (c) (d) (e) (f)

Figure 5. Five-percent salt-and-pepper attack and block-level localization. (a) Watermarked image before attack; (b) image after salt-and-pepper noise attack; (c) ground-truth noise pixel mask; (d) tamper localization overlay; (f) binary tamper localization map.

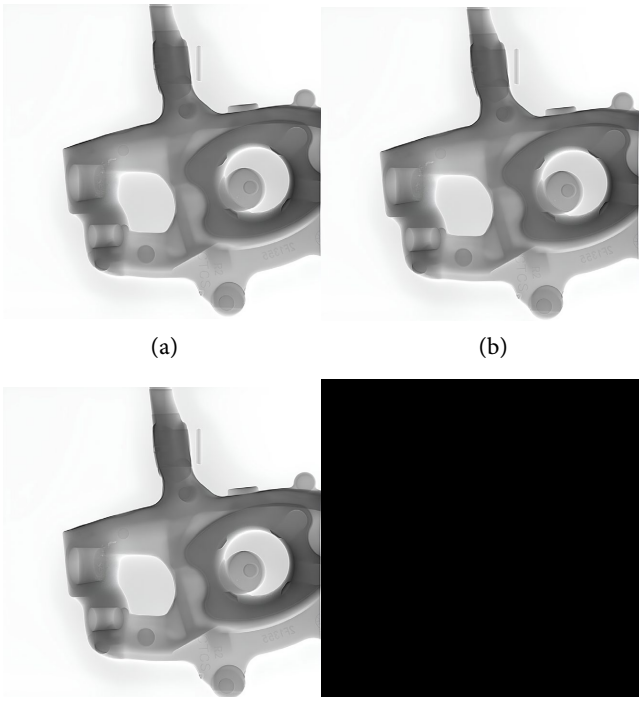
5.3. White-Region Replacement and the No-Tamper Control

With no tampering, attack PSNR is infinite, BER is 0, and NC is 1. The received and pre-attack float64 carriers are identical sample by sample. The confusion counts are TP = 0, FP = 0, TN = 65,536, and FN = 0, indicating no false alarms. Because this control contains no positive block, the interface displays precision, recall, and F1 as zero; these values are not used to evaluate detection performance. Contiguous white-region tampering changes 13.07% of the pixels. The attack PSNR is 15.70 dB, BER is 0.0653, and NC is 0.9865. Localization yields TP = 8697, FP = 0, TN = 56,839, and FN = 0, corresponding to FPR = 0, FNR = 0.00%, precision = 100%, recall = 100.00%, and F1 = 1.0000. Figure 6 shows the original outputs before and after tampering, while Figure 7 and Figure 8 present detection and localization results for the untampered control and region tampering, respectively.



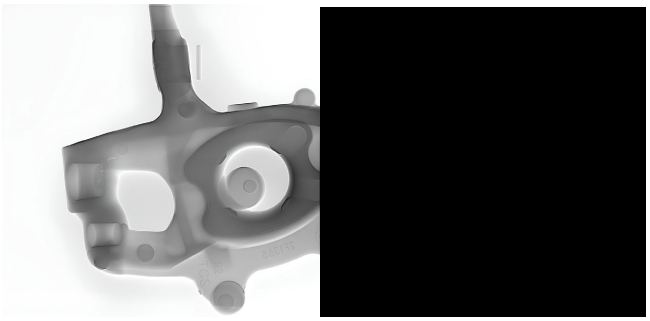
(a) (b)

Figure 6. Watermarked image and the same image after white-region replacement. (a) Watermarked image; (b) white-region replacement.

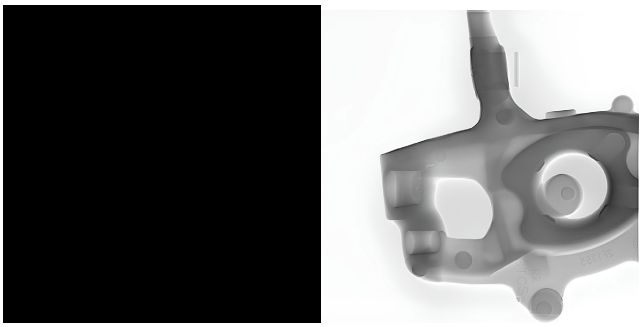


(a) (b)

(c) (d)

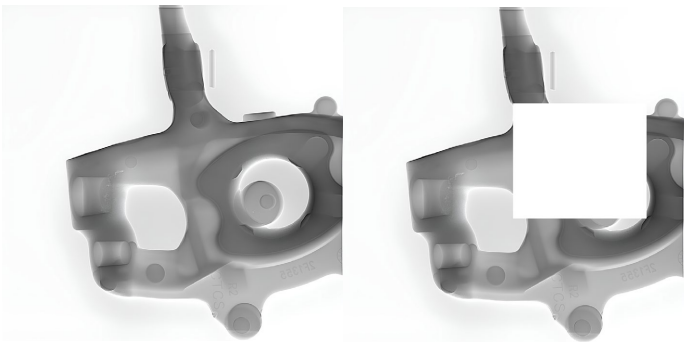


(e) (f)



(g) (h)

Figure 7. Untampered control experiment for cropping attack detection. (a) Original watermarked image; (b) unmodified full image; (c) reconstructed untampered image; (d) ground-truth tamper region mask; (e) tamper localization map; (f) binary tamper localization map; (g) confusion map (TP, FP, FN, TN); (h) tamper localization overlay on the watermarked image.



(a) (b)



(c) (d)

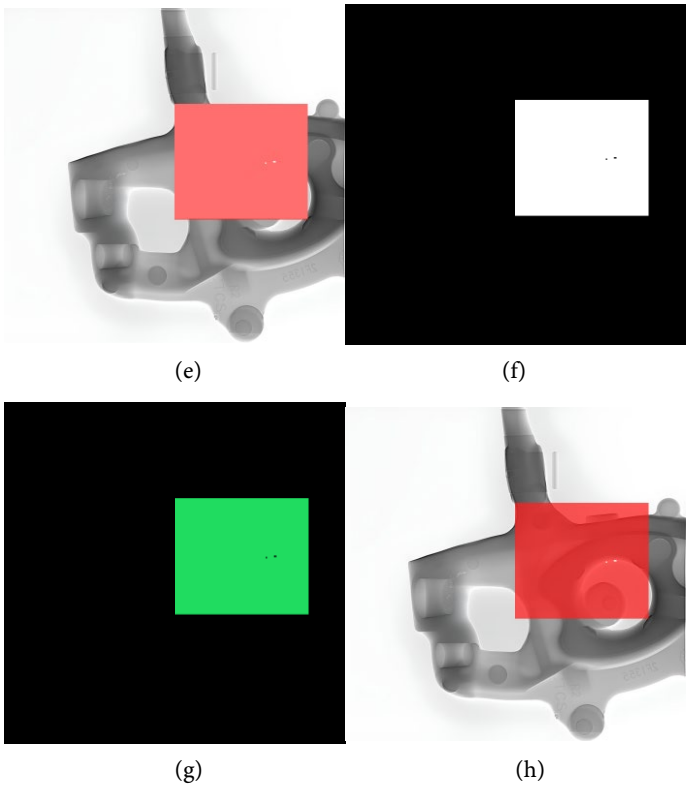


Figure 8. Cropping attack detection and localization experiment with white-region tampering. (a) Original watermarked image; (b) image with white-region tampering; (c) reconstructed attacked image; (d) ground-truth tamper region mask; (e) tamper localization map; (f) binary tamper localization map; (g) confusion map (TP, FP, FN, TN); (h) tamper localization overlay on the watermarked image.

5.4. Supplementary Controlled Experiments

Table 4 reports the target-block decisions. In the untampered control, all 65,536 blocks passed both rules, and the maximum carrier residual was zero. The mean-preserving edit left $D = 0$ but produced a recomputed tag of E0B743 against the extracted tag 9814E2.

Table 4. Target-block decisions in the supplementary experiments.

Condition	Baseline	HMAC tag	Residual δ	Carrier check	Joint rule
No tampering	Pass	Match	0	Pass	Pass
Mean-preserving edit	Pass	Mismatch	0	Pass	Reject
Small carrier perturbation	Pass	Match	1.00×10^{-5}	Reject	Reject

Note. The carrier-comparison tolerance was 10^{-8}.

The perturbation left $D = 0$ and preserved the candidate block and the extracted tag. The measured residual exceeded the tolerance. The joint rule rejected only the target block in both attacks, with zero false alarms elsewhere. Two runs produced identical decisions and residuals.

6. Comparison and Applicability

6.1. Related Fragile-Watermarking Methods

Related methods fall into three groups. The first improves localization reliability by a hierarchy [7], by block features [6], or by dual detection [12]. The second builds and reversibly embeds the authentication data in transform coefficients such as DWT, DCT, or DFT [11] [13]; the slant-transform scheme in [15] provides semi-fragile localization and self-recovery. The third joins authentication to content recovery, copyright protection, or multi-region management, using multi-feature codes [16], hybrid transforms [17], or an ROI/RONI partition [18]. In the experiment reported here, F1 is 1.0000 for the contiguous white region and 0.9989 for 5% salt-and-pepper noise. Table 5 contrasts the mechanisms rather than ranking published numbers, because the image domains and the attack protocols are not the same.

Table 5. Comparison with related fragile-watermarking methods.

Reference	Core mechanism	Main function	Difference from this paper
[13]	Fourier transform and Fibonacci Q-matrix	Reversible authentication	Medical images and a different attack protocol
[14]	Fragile watermark, zero watermark, and EICC chaos	Integrity check, localization, and copyright protection	No content self-recovery
[15]	Slant transform, DNA coding, and chaotic mapping	Localization and self-recovery	Evaluation criteria for semi-fragile authentication and self-recovery differ
[16]	Multi-feature authentication and cross-level embedding	Blind detection and content self-recovery	Uses LBP, DCT, and GLCM features
[17]	Hybrid fragile watermark and dual recovery	Localization and dual self-recovery	Different mapping and recovery mechanism
[18]	Reversible embedding on an ROI/RONI partition	Regional localization and recovery	The boundary region is not fully reversible
This paper	DWT-Logistic-DHT, fixed-range mapping, and retention of anomalous blocks	Block-level authentication and floating-point recovery when there is no attack	Only predefined perturbations were tested; specific blind spots remain

6.2. Applicability and Deployment

Fixed-range mapping acts at feature generation, and retention of anomalous blocks acts after the threshold decision. Both assume that authentication uses the same pixel grid before and after, and that the spatial float64 carrier is kept. An 8-bit preview cannot replace that carrier. JPEG recompression, scaling, and rotation change pixels or block alignment and are not negligible distortions. The baseline results cover the no-tamper case, 5% salt-and-pepper noise, and a contiguous white region. The prototype uses a 24-bit tag to fit the per-block capacity. Applying the guidance in Section 5 of RFC 2104 [19] to HMAC-SHA-256 gives a recommended minimum of 128 bits. The controls on the test image demonstrate how the two checks complement each other. They are not a measure of industrial de-

fect-detection accuracy. The software arranges import, parameter setting, embedding, detection, and export as a sequence that can be inspected, and it stores the high-precision NPZ file separately from the 8-bit preview. Figures 1-3 record the watermarking workflow, Figure 4 previews the industrial casting X-ray image collection, and Figures 5-8 record attack evaluation. Those views correspond to the measures in Table 2 and Table 3. Table 5 places the baseline against related designs.

7. Conclusion

Industrial X-ray images must keep their original pixels through storage and transmission and must still reveal local edits. Block-mean authentication with a fixed-range mapping reduces the cross-block false alarms of image-level normalization, but a feature that is only a mean cannot see a replacement that preserves the mean. This paper therefore proposes block-level authentication that joins a complete-block HMAC with a carrier-consistency check. The code covers every pixel of the 4×4 block and is bound to the image identity and the block location. Authentication bits are embedded in the fractional part of selected DHT coefficients. The receiver checks each block code and compares the received carrier with the carrier rebuilt from the recovered candidate and the code. The two checks target different ways in which a mean feature can fail. On the test image, the HMAC check detected the mean-preserving edit and the carrier check detected the small perturbation. The joint rule localized the target block in both attacks without false alarms in these single-image tests of the 24-bit prototype. Future work will increase the number and diversity of industrial X-ray images and add tests involving mean-preserving detail edits, small-region replacements, and small carrier perturbations to examine the detection and localization performance of the proposed authentication approach under different image and tampering conditions. Detection results, localization accuracy, and image-recovery accuracy will then be compared across block sizes and DHT embedding configurations to analyze the effects of parameter settings. Running time, memory consumption, and high-precision carrier storage will also be recorded for each configuration to inform parameter selection in practical applications.

Acknowledgements

This work was supported by the Key Research and Development Program of Shaanxi Province (Grant No. 2025GY-YBXM-068), "Key Technologies of Digital Copyright Protection Based on Hypercomplex Fractional-order Image Moments and Machine Learning", and by the Key Scientific Research Program of the Shaanxi Provincial Department of Education (Grant No. 25JR067), "Industrial-Image Defect Detection Based on Image Moments and Lightweight Neural Networks".

Author Contributions

Conceptualization, Peichen Zheng; methodology, Peichen Zheng; software, Shijie

Wang; validation, Peichen Zheng; formal analysis, Shugang Cheng; investigation, Gengwu Chen; data curation, Jing Wang; writing—original draft preparation, Peichen Zheng and Shijie Wang; writing—review and editing, Shugang Cheng; visualization, Gengwu Chen; supervision, Bing HE; project administration, Bing HE; funding acquisition, Bing HE. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Ou, B., Yin, Z.X. and Xiang, S.J. (2022) A Survey of Reversible Information Hiding in Plaintext Images. *Journal of Image and Graphics*, **27**, 111-124. (In Chinese)

[2] Tian, J. (2003) Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, **13**, 890-896. https://doi.org/10.1109/tcsvt.2003.815962

[3] Wen, J.F., Wang, J.Z., Liu, A.Z., *et al.* (2009) Reversible Image Authentication Based on Difference Expansion and Error-Correcting Codes. *Computer Engineering*, **35**, 125-127. (In Chinese)

[4] Zhong, Y.Y. and Huang, F.J. (2023) An Efficient Block-Based Reversible Image Authentication Method. *Journal of Software*, **34**, 5848-5861. (In Chinese)

[5] Fan, X.H. and Li, H. (2019) Reversible Data Hiding for Medical Images Based on Prediction Error. *Journal of Beijing University of Chemical Technology* (*Natural Science Edition*), **46**, 83-89. (In Chinese)

[6] Zhang, J.P., Zhang, Q.F. and Yang, H.J. (2014) Image Tamper Detection and Recovery Based on Block Features and Chaotic Sequences. *Journal of Shandong University* (*Engineering Science*), **44**, 63-69. (In Chinese)

[7] Celik, M.U., Sharma, G., Saber, E., *et al.* (2002) Hierarchical Watermarking for Secure Image Authentication with Localization. *IEEE Transactions on Image Processing*, **11**, 585-595. https://doi.org/10.1109/tip.2002.1014990

[8] He, H.J. and Chen, F. (2007) Security Analysis of Self-Embedding Watermarking Algorithms. *Acta Electronica Sinica*, **35**, 557-562. (In Chinese)

[9] Evsutin, O., Melman, A. and Meshcheryakov, R. (2020) Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions. *IEEE Access*, **8**, 166589-166611. https://doi.org/10.1109/access.2020.3022779

[10] Shehab, A., Elhoseny, M., Muhammad, K., Sangaiah, A.K., Yang, P., Huang, H., *et al.* (2018) Secure and Robust Fragile Watermarking Scheme for Medical Images. *IEEE Access*, **6**, 10269-10278. https://doi.org/10.1109/access.2018.2799240

[11] Bouarroudj, R., Souami, F., Bellala, F.Z., Zerrouki, N., Harrou, F. and Sun, Y. (2025) Secure and Reversible Fragile Watermarking for Accurate Authentication and Tamper Localization in Medical Images. *Computers and Electrical Engineering*, **123**, Article 110072. https://doi.org/10.1016/j.compeleceng.2025.110072

[12] Zhan, C., Leng, L., Chang, C. and Horng, J. (2024) Reversible Image Fragile Watermarking with Dual Tampering Detection. *Electronics*, **13**, Article 1884. https://doi.org/10.3390/electronics13101884

[13] Bouarroudj, R., Souami, F., Zohra Bellala, F. and Zerrouki, N. (2024) A Reversible

Fragile Watermarking Technique Using Fourier Transform and Fibonacci Q-Matrix for Medical Image Authentication. *Biomedical Signal Processing and Control*, **92**, Article 105967. https://doi.org/10.1016/j.bspc.2024.105967

[14] Yan, F., Wang, Z. and Hirota, K. (2024) Dual Medical Image Watermarking Using SRU-Enhanced Network and EICC Chaotic Map. *Complex & Intelligent Systems*, **11**, Article No. 101. https://doi.org/10.1007/s40747-024-01723-6

[15] Jagadeesh, S., Parida, A.K. and Meenakshi, K. (2025) Tamper Detection, Localization and Self-Recovery Using Slant Transform with DNA Encoding for Medical Images. *Results in Engineering*, **27**, Article 105907. https://doi.org/10.1016/j.rineng.2025.105907

[16] Wu, Q., Li, H., Li, M. and Wang, M. (2026) Multi-Feature Fragile Image Watermarking Algorithm for Tampering Blind-Detection and Content Self-Recovery. *Computers*, *Materials & Continua*, **86**, 1-20. https://doi.org/10.32604/cmc.2025.068220

[17] Sahu, A.K. and Sahu, M. (2026) Hybrid Fragile Image Watermarking for Tamper Detection, Localization and Dual Self-Recovery. *Engineering Science and Technology*, **73**, Article 102266. https://doi.org/10.1016/j.jestch.2025.102266

[18] Arham, A., Adji, T.B., Alfarozi, S.A.I. and Nugroho, H.A. (2026) Reversible Watermarking for Medical Images Using Roi-Based Tamper Localization and Recovery. *Franklin Open*, **15**, Article 100545. https://doi.org/10.1016/j.fraope.2026.100545

[19] Krawczyk, H., Bellare, M. and Canetti, R. (1997) HMAC: Keyed-Hashing for Message Authentication. RFC 2104. https://doi.org/10.17487/RFC2104