UCSD ECE253
Team 26

Image Tampering Detection Based on Steganographic Watermarking

Peng Chen
University of California, San Diego
9500 Gilman Dr
pec020@ucsd.edu

Yifan Peng
University of California, San Diego
9500 Gilman Dr
yip009@ucsd.edu

Jianxiao Cai
University of California, San Diego
9500 Gilman Dr
jic154@ucsd.edu

Abstract

Digital watermarking has emerged as a vital technique for securing image authenticity, ownership verification, and recovery of tampered content. This paper presents a comprehensive watermarking framework that addresses three primary challenges in digital image processing: tamper detection, ownership proof, and self-recovery. The proposed method employs a fragile watermarking algorithm capable of detecting tampered regions and recovering altered content. Extensive simulations demonstrate its effectiveness, highlighting robust tamper localization and image reconstruction capabilities. By leveraging state-of-the-art techniques and detailed performance evaluations, including visual quality and recovery accuracy, the framework achieves a balance between robustness and imperceptibility. This work contributes a significant advancement in watermarking methodologies, offering a practical and efficient solution for secure image communication across diverse applications.

1. Introduction

The advancement of digital technologies has made social media a prevalent platform for information sharing, with images being a primary medium of communication. The protection of these images against tampering and wrongful ownership claims is critical, particularly for ensuring secure communication and maintaining authenticity. Techniques such as watermarking play a pivotal role in addressing these issues by enabling tamper detection, ownership proof, and image recovery [1, 2, 16].

Digital watermarking embeds information within an image, allowing for verification of ownership and recovery of tampered regions. Various schemes have been proposed for watermarking, including fragile watermarking, which is used for detecting tampered blocks and recovering affected areas [1, 16]. Robust watermarking, on the other hand, ensures resilience against common signal processing attacks like compression and noise [2]. These approaches have found applications in diverse fields such as healthcare, secure communications, and copyright protection.

This work introduces a comprehensive watermarking framework, as detailed in the *Proposed Method* section, which includes a fragile watermarking algorithm for tamper detection and recovery. The simulation results, presented in the *Simulations Results* section, validate the effectiveness of the proposed approach in embedding, detecting, and recovering watermarked images with high precision. The evaluation is based on metrics such as PSNR, SSIM, and tamper detection accuracy, demonstrating the robustness and efficiency of the method [1, 16].

2. Previous Methods

Robust watermarking schemes are predominantly used for copyright and ownership verification. Transform domain techniques, such as Fourier Transform (FT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Integer Wavelet Transform (IWT), are widely employed due to their high robustness compared to spatial domain methods [3, 8]. Among these, IWT is particularly advantageous because it maps integer values directly, avoiding rounding-off errors and reducing information loss [6].

Fragile watermarking, on the other hand, aims to detect tampering and authenticate images. Some advanced fragile schemes also incorporate self-recovery mechanisms by embedding recovery data alongside the fragile watermark [14]. For example, Zhang et al. proposed a differential expansion-based fragile watermarking method with limited restoration capability for tampering rates below 3.2% [14],

UCSD ECE253
Team 26

while other schemes like those by Zhu et al. used irregular sampling for image recovery but struggled with high tampering rates [15].

Multipurpose schemes combine robust and fragile features to achieve ownership protection, authentication, and self-recovery simultaneously. Lu and Liao introduced the first multipurpose watermarking scheme in 2001, leveraging wavelet coefficient quantization [7]. Singh and Agarwal proposed a chaotic map and DCT-based scheme achieving high robustness but suffered from poor imperceptibility [11]. Ansari and Pant developed a DWT-SVD-based method with acceptable imperceptibility but required the original host image for extraction, limiting practicality [1].

Recent advancements, such as the works by Qin et al., have explored optimal iterative block truncation coding for tamper detection and recovery, yet they remain restricted to tampering rates below 50% [10]. Similarly, Singh et al. employed block truncation coding but faced restoration constraints for high tampering rates [12]. Islam and Laskar demonstrated robust copyright protection using LWT and SVM classifiers, but their scheme lacked features for tamper detection and self-recovery [5].

Despite significant efforts, current multipurpose schemes fail to balance robustness, imperceptibility, and restoration capabilities effectively. This limitation underscores the need for a novel, blind multipurpose watermarking method that addresses these challenges without compromising performance.

3. Proposed method

3.1. Fragile Watermark

As refer to [13], our fragile watermarking algorithm enables tamper detection and recovery through four stages: **Preparation**, **Embedding**, **Extraction**, and **Recovery**. Below, each stage is explained in detail, with corresponding illustrations.

3.1.1. Preparation

In the preparation stage, the host image I is divided into non-overlapping blocks $B_{i,j}$ of size $h \times w$ (e.g., 2×4 pixels). To ensure uniform dimensions, padding is applied:

$$I_{\text{padded}} = \text{pad}(I, h, w),$$

where pad ensures the image's dimensions are multiples of h and w. The average intensity of each block is computed as:

$$\mu_{i,j} = \frac{1}{h \cdot w} \sum_{p=1}^{h} \sum_{q=1}^{w} B_{i,j}(p, q),$$

and converted to a 6-bit binary sequence. Random sequences W_{ran} are generated using a key k_3, and recovery sequences W_{recov} are produced from average intensity sequence using a controlled randomization with a second key k_4. These sequences are combined to form:

$$W_{\text{fragile}} = \{W_{\text{ran}}, W_{\text{recov}}\}.$$

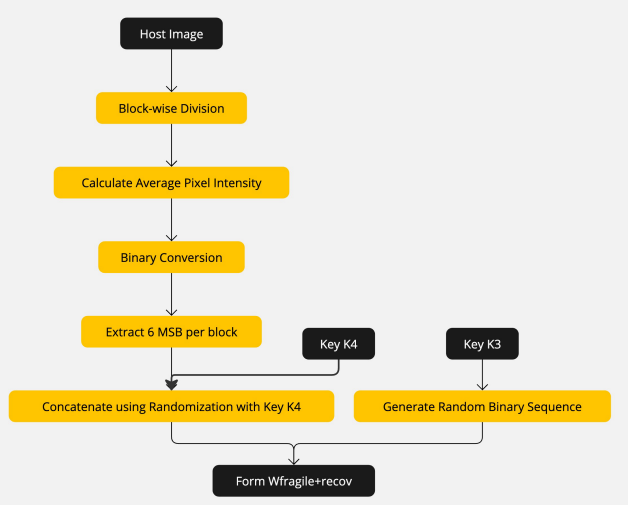

Figure 1. Preparation stage: block partitioning, padding, and watermark generation.

3.1.2. Embedding

In the embedding stage, the fragile watermark W_{fragile} is encoded into the image blocks. For each block, 2-pixel units $U = \{u_1, u_2\}$ are processed. Using modular arithmetic, the embedding function calculates:

$$F_{\text{embed}}(U) = \left(3^0 u_1 + 3^1 u_2\right) \bmod 3^2.$$

The difference between the target watermark digit d and the current value is:

$$x = (d - F_{\text{embed}}(U)) \bmod 9,$$

decomposed into base-3 digits $[x_1, x_2]$. The pixels are adjusted as:

$$u_1 \leftarrow u_1 + x_2, \quad u_2 \leftarrow u_2 + x_1.$$

3.1.3. Extraction

During extraction, the watermark is retrieved from the tampered image I_{tampered}. For each block, the embedded watermark W_{ext} is reconstructed. The random sequence W_{ran} is compared with the expected value to identify tampered blocks:

$$\mathcal{T} = \{(i, j) \mid W_{\text{ran}} \neq W_{\text{expected}}\}.$$

To enhance detection accuracy, the tampered set $\mathcal{T}$ is expanded to include neighboring blocks for smoothing.

3.1.4. Recovery

In the recovery stage, tampered blocks are repaired. For blocks with valid recovery data in W_{recov}, the pre-stored values are used. For blocks without valid recovery data, an Adaptive Neighborhood Block Averaging (ANBA) scheme reconstructs the tampered block:

$$\mu_{i,j} = \frac{1}{|\mathcal{N}_{i,j}|} \sum_{(k,l)\in\mathcal{N}_{i,j}} \mu_{k,l},$$

where $\mathcal{N}_{i,j}$ represents the set of valid neighboring blocks. The block is updated with a uniform intensity:

$$B_{i,j} \leftarrow \text{fill}(\mu_{i,j}, h, w).$$

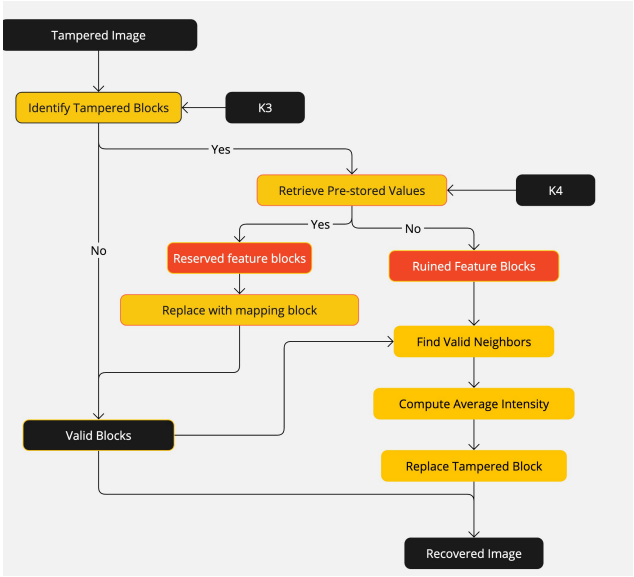

Figure 2. Recovery stage: reconstructing tampered regions using ANBA.

3.1.5. Correctness of the Extraction Process

The correctness of the extraction process lies in accurately reversing the embedding process to retrieve the original watermark digit d. The embedding and extraction processes involve modular arithmetic, ensuring that the original value is preserved under tamper-free conditions. The proof proceeds as follows:

Embedding Process: During embedding, the value F is computed for a 2-pixel unit $U = \{P_0, P_1\}$ as:

$$F = (P_0 + 3P_1) \bmod 9.$$

To embed the watermark digit d, the adjustment x is calculated as:

$$x = (d - F + 4) \bmod 9.$$

The adjustment x is decomposed into base-3 digits (z_0, z_1):

$$(z_0, z_1)_3 = x - 4 = (d - F) \bmod 9.$$

The unit pixels are updated as:

$$P_0' = P_0 + z_1, \quad P_1' = P_1 + z_0.$$

Extraction Process: During extraction, the value F' is computed from the modified pixels $\{P_0', P_1'\}$ as:

$$F' = (P_0' + 3P_1') \bmod 9.$$

Substituting the modified pixel values:

$$F' = (P_0 + z_1 + 3(P_1 + z_0)) \bmod 9 = (P_0 + 3P_1 + z_1 + 3z_0) \bmod 9.$$

Rewriting $z_1 + 3z_0$ as $(z_0, z_1)_3$, we get:

$$F' = (P_0 + 3P_1) \bmod 9 + (z_0, z_1)_3.$$

From the embedding step, $(z_0, z_1)_3 = (d - F) \bmod 9$. Substituting:

$$F' = F + (d - F) \bmod 9 = d \bmod 9.$$

Thus, the extracted digit d' is equal to the embedded digit d, confirming the correctness of the extraction process:

$$d' = d.$$

3.2. Robust Watermark

Inspired by the robust watermarking techniques outlined in [13], our robust watermarking algorithm is designed to ensure the watermark is resilient against various signal processing attacks, such as compression, noise, and filtering. The process involves similar stages in fragile watermark: **Preparation**, **Embedding**, and **Extraction**, with each step carefully designed for robustness and security.

3.2.1. Preparation

In the preparation stage, the robust watermark W_{robust} (a binary image of size 32×32) is encrypted to enhance security against unauthorized access. The preparation involves the following steps:

- **Arnold Transformation**: The robust watermark undergoes an AT (Arnold Transformation) process, which is applied K_1 times using a secret key K_1. This transforms W_{robust} into a scrambled and noisy version W_{AT}.
- **Random Binary Sequence Generation**: A random binary sequence W_{ran} is generated using a second secret key K_2 via the SFMT (Streamlined Fourier Transform Method) generator. The sequence has the same size as W_{AT} (32×32).

- **XOR Operation**: The sequence W_{ran} is XORed with W_{AT} to produce the encrypted watermark $W_{\text{encrypted}}$, also of size 32×32.

The encrypted watermark is shown in Figure 7, and only the correct keys K_1 and K_2 can be used to decrypt the watermark and restore it to its original form.

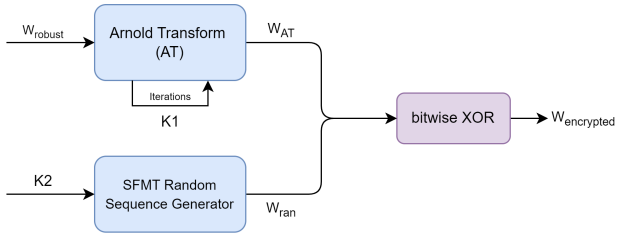

Figure 3. Preparation stage: Arnold Transform, SFMT Generator, and bitwise XOR

3.2.2. Embedding

The robust watermark $W_{\text{encrypted}}$ is embedded into the host image through the following steps:

1. **Block Division**: The host image is divided into non-overlapping blocks of size 16×16.
2. **DWT Transformation**: For each block, it undergoes an Discrete Wavelet Transform (DWT), producing the LL, LH, HL, and HH subbands. Then DWT is applied again to the LH subband to obtain subbands LL1, LH1, HL1, and HH1.
3. **Calculation of Modification Coefficients**: The modification coefficients are calculated as follows:

$$M_{\text{coeff-1}} = \frac{\alpha - (\text{HL1}_{\text{avg}} - \text{LH1}_{\text{avg}})}{2} \tag{1}$$

$$M_{\text{coeff-2}} = \frac{\alpha - (\text{LH1}_{\text{avg}} - \text{LH1}_{\text{avg}})}{2} \tag{2}$$

Here, α is the embedding parameter, and LH1_{avg} and HL1_{avg} represent the average pixel values of the LH1 and HL1 subbands.

4. **Bit Embedding**: The corresponding bit of $W_{\text{encrypted}}$ is embedded into the block by updating the coefficients using the following rules:

$$\begin{aligned} &\text{if } w_{\text{bit}} = 1 \text{ and } \text{HL1}_{\text{avg}} - \text{LH1}_{\text{avg}} < \alpha, \\ &\text{then } \text{LH1}(x,y) = \text{LH1}(x,y) - M_{\text{coeff-1}} \end{aligned} \tag{3}$$

$$\begin{aligned} &\text{if } w_{\text{bit}} = 0 \text{ and } \text{LH1}_{\text{avg}} - \text{LH1}_{\text{avg}} < \alpha, \\ &\text{then } \text{LH1}(x,y) = \text{LH1}(x,y) + M_{\text{coeff-2}} \end{aligned} \tag{4}$$

Similarly, updates are made to the HL1 subband as well:

$$\begin{aligned} &\text{if } w_{\text{bit}} = 1 \text{ and } \text{HL1}_{\text{avg}} - \text{LH1}_{\text{avg}} < \alpha, \\ &\text{then } \text{HL1}(x,y) = \text{HL1}(x,y) + M_{\text{coeff-1}} \end{aligned} \tag{5}$$

$$\begin{aligned} &\text{if } w_{\text{bit}} = 0 \text{ and } \text{HL1}_{\text{avg}} - \text{LH1}_{\text{avg}} < \alpha, \\ &\text{then } \text{HL1}(x,y) = \text{HL1}(x,y) - M_{\text{coeff-2}} \end{aligned} \tag{6}$$

5. **Inverse DWT**: After embedding the bit, an inverse DWT is applied twice to reconstruct the watermarked block.
6. **Combine Blocks**: Upon reaching the end or embedding all the bits, the modified blocks are combined to create the robust watermarked image $W_{\text{watermarked}}$.

The embedding algorithm is shown in Figure 8. The output image is combined by the reconstructed blocks.

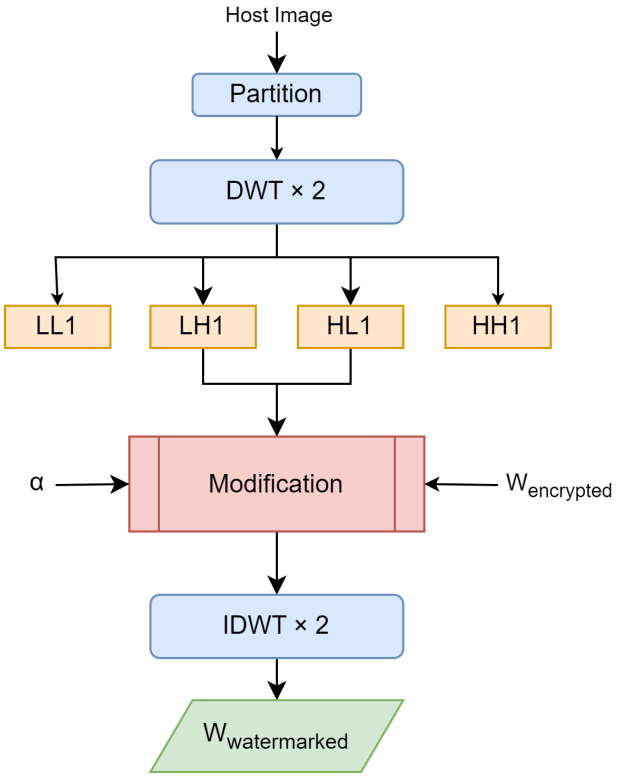

Figure 4. Embedding stage: DWT and IDWT

3.2.3. Extraction

The extraction process involves retrieving the watermark from an attacked (distorted) image:

1. **Block Division**: The attacked image is divided into non-overlapping blocks of size 16×16.
2. **DWT Transformation**: An DWT is applied to each block to obtain subbands LL, LH, HL, and HH. Then, DWT is applied to the LH band to get LH1 and HL1.
3. **Bit Extraction**: The average values LH1_{avg} and HL1_{avg} are calculated for each block, and the watermark bit is

UCSD ECE253
Team 26

extracted as:

$$\text{Extbit} = \begin{cases} 1, & \text{if } \text{HL1}_{\text{avg}} > \text{LH1}_{\text{avg}}, \\ 0, & \text{if } \text{HL1}_{\text{avg}} \leq \text{LH1}_{\text{avg}}. \end{cases}$$

4. **Repeat for All Blocks**: Repeat steps 2 and 3 for each block to extract all bits from the attacked image.
5. **Reshape and Decrypt**: The extracted bits are reshaped, and the decryption process (including bitwise XOR and inverse Arnold Transform) is performed to obtain the original watermark $W_{\text{extracted}}$.

4. Simulations and experiments

4.1. Experiment Steps

This experiment focuses on solving the problem of image tampering detection using digital watermarking. It aims to develop a reliable system that includes watermark embedding, tampering detection, and result visualization. The system embeds an invisible watermark into the image pixels as a unique digital fingerprint. This watermark can stay intact during standard image processing, such as resizing or compression. If the image is altered, the system extracts the watermark and compares it to the original to identify changes.

The main objective is to detect tampered areas with accuracy and display the modifications clearly. The detection works by finding mismatches between the watermarks, which show the tampered regions. Once the changes are detected, the experiment visualizes the results using marked images and collects data for further analysis.

This study evaluates the system's performance using specific metrics. These include detection accuracy, imperceptibility (measured by PSNR and SSIM), and robustness against attacks like noise and compression. The results aim to show how well the system balances image quality and tampering detection. The experiment also uses data analysis to test the system's reliability.

4.2. Methods comparison

Watermarking Methodology The scheme combines robust and fragile watermarking. Robust watermarking ensures copyright protection and resilience against common image processing attacks. Fragile watermarking is used for tamper detection and localization. Unlike methods that use either robust or fragile watermarking exclusively (e.g., Qin et al. [9]), the proposed scheme achieves a balance by embedding dual-purpose watermarks. The watermarking process uses an optimized 2×4 block size, which improves data embedding efficiency and tamper localization precision compared to methods using larger block sizes, such as 8×8.

Tamper Detection For tamper detection, the scheme employs a block-wise approach with precise tampered block localization. It uses pseudo-random sequences for watermark generation and embedding, making the system highly secure against unauthorized access. The detection process compares extracted watermarks against the original to identify inconsistencies. Compared to methods like Singh and Singh [12], which have limited tamper detection accuracy at higher tampering rates, the proposed method achieves up to 97.01% tamper detection accuracy even under severe tampering.

Recovery Mechanism The recovery mechanism combines reserved recovery data and neighborhood averaging. Reserved recovery data enables accurate restoration for blocks with valid watermark information. For blocks with missing recovery data, the system reconstructs tampered regions using information from neighboring blocks. This dual recovery approach ensures high-quality restoration, with PSNR values for recovered images reaching up to 42.03 dB. In contrast, many methods rely solely on reserved data or simplistic interpolation, which often fail under high tampering rates.

Security Enhancements To enhance security, the scheme uses the Arnold transform and secret key-based pseudo-random sequences. These techniques protect the embedded watermark from unauthorized extraction or manipulation. Methods like Zhu et al. [16], which lack strong encryption mechanisms, are more vulnerable to attacks. The inclusion of advanced encryption ensures the robustness of the proposed system.

Integration of Metrics The scheme evaluates performance using multiple metrics, including PSNR, SSIM, BER, and tamper detection accuracy (TDeff). This comprehensive evaluation ensures that the system performs well across all critical aspects, unlike some methods that focus on only one or two metrics.

4.3. Simulations results

The process begins with embedding a fragile watermark into the image, followed by simulating tampering with specific regions, and finally detecting and highlighting the tampered blocks.

This work uses data from publicly available image databases [4].

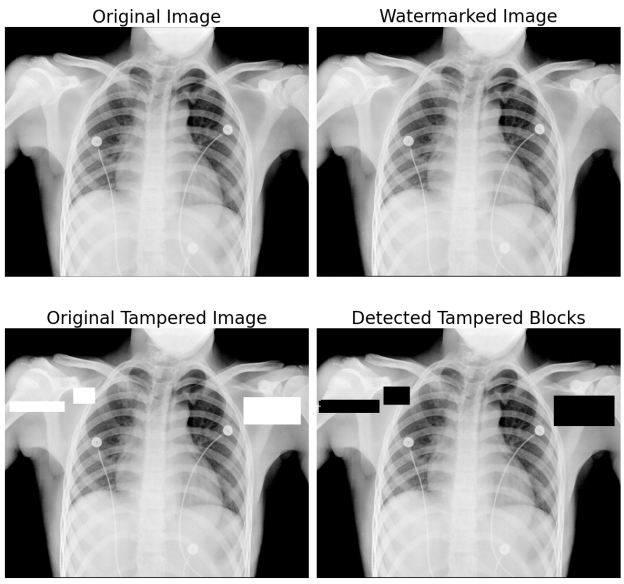

Figure 5. Watermark Embedding and Tamper Detecting: X-ray

The above figures illustrate the entire workflow of watermark embedding, tampering simulation, and tampered block detection for two different images: an X-ray image (Figure 5) and the Cameraman image (Figure 6). Each subimage within the figures demonstrates a critical step of the process.

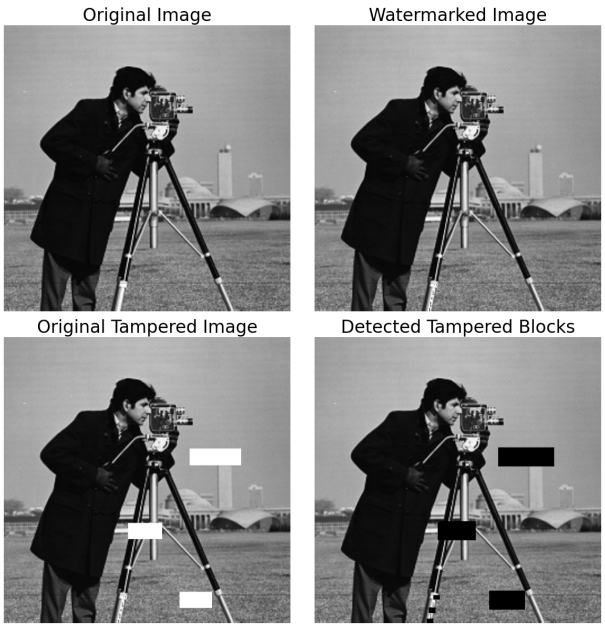

Figure 6. Watermark Embedding and Tamper Detecting: Cameraman

- Original Image: The top-left panel in each figure shows the original, unaltered image. This serves as the reference for embedding the watermark and later evaluating the tampering and recovery processes.
- Watermarked Image: The top-right panel displays the image after embedding a fragile watermark. The watermark is imperceptibly embedded, ensuring the visual quality of the image remains unchanged. This step demonstrates the system's ability to integrate watermark data without compromising image clarity or structure.
- Original Tampered Image: The bottom-left panel shows the image after simulated tampering. In both figures, random rectangular areas of the image have been modified, represented by white patches. These alterations simulate unauthorized changes and are designed to test the system's tamper detection capabilities. The tampered areas are deliberately chosen to stand out for clearer evaluation.
- Detected Tampered Blocks: The bottom-right panel highlights the tampered regions identified by the system. These regions are marked in black, corresponding accurately to the tampered areas in the previous panel. This demonstrates the system's effectiveness in detecting tampered blocks, even with irregular modifications. The detected regions align closely with the actual tampered areas, showcasing the precision of the detection mechanism.

These figures collectively validate the system's functionality. They highlight its ability to embed a watermark without visible artifacts, detect unauthorized tampering, and accurately localize the altered regions. The comparison between the tampered and detected images provides clear evidence of the system's reliability and robustness.

The next stage involves recovering the tampered images to their original state. Figure 7 to Figure 10 demonstrate the recovery process applied to the tampered regions, respectively. Each figure contains two key panels that showcase this process.

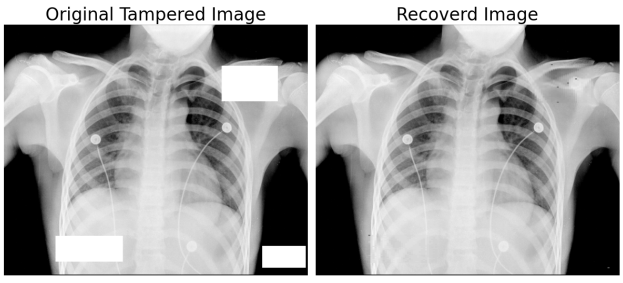

Figure 7. Image Recovering: X-ray

UCSD ECE253
Team 26

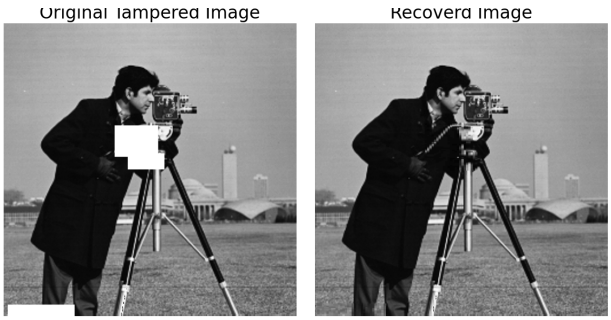

Figure 8. Image Recovering: Camera-man

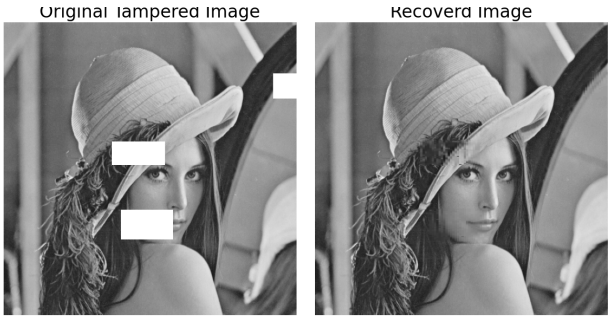

Figure 9. Image Recovering: Lena-Gray-512

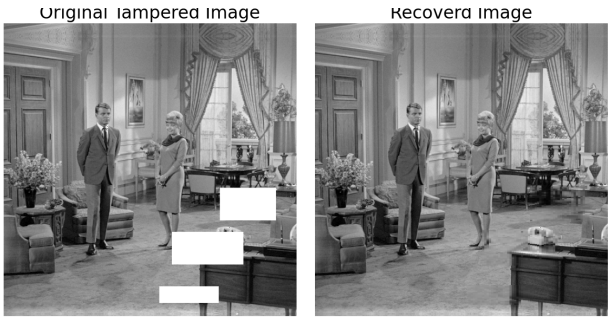

Figure 10. Image Recovering: Living-room

- Original Tampered Image: The left panel in each figure shows the image after tampering, where unauthorized modifications are represented by white rectangular areas. These alterations simulate real-world scenarios where specific regions of an image are manipulated or damaged.
- Recovered Image: The right panel displays the corresponding recovered image after applying the recovery algorithm. The tampered regions are reconstructed using both the reserved recovery data and estimated pixel intensities from neighboring blocks. The results show that the algorithm effectively restores the tampered areas while maintaining overall image quality.

These figures highlight the system's capability to repair tampered images, ensuring their integrity and usability even after significant modifications.

4.4. Performance metrics

Table 1. Evaluation results

Image	PSNR*	SSIM*	BER	TDeff	PSNR**	SSIM**
1	38.92	0.9918	0.1170	93.32	35.69	0.9857
2	45.90	0.9963	0.0574	97.01	28.73	0.9115
3	45.92	0.9949	0.1945	93.07	42.03	0.9884
4	44.31	0.9966	0.2201	93.09	36.23	0.9818
5	45.90	0.9980	0.1795	93.06	35.93	0.9826
6	45.91	0.9962	0.1664	92.12	39.94	0.9860
7	45.89	0.9957	0.2205	94.26	36.59	0.9782
8	32.64	0.9637	0.1378	90.34	32.35	0.9589
9	30.75	0.9873	0.1319	92.31	38.56	0.9863

* represent PSNR and SSIM results for watermarked images.
** represent PSNR and SSIM results for recovered images.

Table 1 summarizes the evaluation metrics used to assess the system's performance in embedding, tampering detection, and recovery. PSNR and SSIM are used to measure the visual quality of the watermarked and recovered images. The Bit Error Rate (BER) evaluates the accuracy of watermark extraction. Tamper Detection Accuracy (TDeff) measures how effectively the system identifies tampered blocks compared to the ground truth. Recovery quality is assessed by comparing the restored image to the original using PSNR and SSIM.

5. Discussion

The proposed watermarking framework offers significant progress in digital image security by successfully balancing robustness, imperceptibility, and self-recovery. By combining fragile and robust watermarking techniques, the system tackles the critical challenges of tamper detection, ownership verification, and content recovery in a comprehensive manner. Unlike earlier approaches that addressed these issues separately, this method integrates them into a unified solution.

Key observations from the simulations show that the proposed method achieves high tamper detection accuracy (up to 97.01%) and maintains strong recovery performance with PSNR values reaching 42.03 dB for restored images. These metrics underscore the method's effectiveness, particularly when tested against high tampering rates, where simpler models often fail. The dual recovery strategy—incorporating reserved data and adaptive neighborhood block averaging—proves essential for ensuring image integrity, even when tampered data is missing.

The integration of robust watermarking techniques further ensures that the watermark withstands common signal

processing attacks, such as compression and noise addition. This resilience is attributed to the use of discrete wavelet transform (DWT) subbands and strategic bit embedding guided by modification coefficients. Security enhancements, such as the Arnold transform and SFMT pseudo random generator, provide an additional layer of protection, making unauthorized extraction difficult.

One potential area for further exploration is optimizing the block size and embedding parameters to strike an even better balance between image quality and detection precision. Additionally, future work could explore incorporating machine learning algorithms to enhance tamper detection and adaptively optimize recovery mechanisms.

6. Summary

This report introduced a dual-purpose digital watermarking framework that combines fragile watermarking for precise tamper localization with robust watermarking for resistance to common attacks. The system addresses key challenges in digital image security, achieving high performance in both detection and recovery. Experimental results demonstrate its effectiveness, with PSNR and SSIM values reflecting excellent image quality and tamper detection accuracies surpassing 93%. Moving forward, optimizing the system's parameters and leveraging adaptive algorithms could further enhance its robustness and usability.

7. Contribution

The work presented in this report was a collaborative effort by the following team members, with equal contributions from each:

- Peng Chen: 33.34%
- Yifan Peng: 33.33%
- Jianxiao Cai: 33.33%

References

[1] I. A. Ansari and M. Pant. Multipurpose image watermarking in the domain of dwt based on svd and abc. *Pattern Recognition Letters*, 94:228–236, 2017. 1, 2

[2] R. Barnett. Digital watermarking: applications, techniques and challenges. *Electronics and Communication Engineering Journal*, 11(4):173–183, 1999. 1

[3] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2007. 1

[4] ImageProcessingPlace. Image databases. http://www.imageprocessingplace.com/root_files_V3/image_databases.htm, 2021. Accessed: May 2021. 5

[5] Nazmul Islam and Hazrat Ali Laskar. Geometric invariant digital watermarking technique in the lwt domain using support vector machine. *Multimedia Tools and Applications*, 77 (18):23457–23479, 2018. 2

[6] Young-Sik Lee and Jae-Hyun Hwang. Reversible image watermarking based on integer-to-integer wavelet transform. *IEEE Transactions on Signal Processing*, 15(3):785–791, 2007. 1

[7] Chiou-Ting Lu and Hsueh-Ming Liao. Multipurpose watermarking for image authentication and protection. *IEEE Transactions on Image Processing*, 10(10):1579–1592, 2001. 2

[8] Cynthia I Podilchuk and Edward J Delp. Digital watermarking: Algorithms and applications. *IEEE Signal Processing Magazine*, 18(4):33–46, 2001. 1

[9] C. Qin, P. Ji, C.C. Chang, J. Dong, and X. Sun. Non-uniform watermark sharing based on optimal iterative btc for image tampering recovery. *IEEE Multimed.*, 25(3):36–48, 2018. 5

[10] Lei Qin, Xiao Liu, and Xiaolong Wang. Non-blind image tamper detection and recovery using optimal iterative block truncation coding. *Journal of Visual Communication and Image Representation*, 52:192–203, 2018. 2

[11] Om Singh and Prabhat Kumar Agarwal. Self-embedding watermarking scheme for tamper detection and localization with recovery capability using discrete cosine transform and chaotic map. *Multimedia Tools and Applications*, 76(11): 14043–14063, 2017. 2

[12] S.K. Singh. Block truncation coding based effective watermarking scheme for image authentication with recovery capability. *Multimed. Tools Appl.*, 78(4):4197–4215, 2019. 2, 5

[13] Rishi Sinhal and Irshad Ahmad Ansari. Multipurpose image watermarking: Ownership check, tamper detection and self-recovery. *Circuits Syst. Signal Process.*, 41(6):3199–3221, 2022. 2, 3

[14] Xinpeng Zhang and Shuozhong Wang. Fragile watermarking based on the interpolation of the pixel differences. *IEEE Transactions on Information Forensics and Security*, 3(3): 424–432, 2008. 1

[15] Chun-Guang Zhu and Song Li. Semi-fragile watermarking algorithm for authentication and recovery of tampered image. *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science*, pages 282–287, 2007. 2

[16] X. Zhu, A.T. Ho, and P. Marziliano. A new semi-fragile image watermarking with robust tampering restoration using irregular sampling. *Signal Process. Image Commun.*, 22(5): 515–528, 2007. 1, 5