Highlights

An Efficient Digital Watermarking Technique for Small Scale devices

Kaushik Talathi, Aparna Santra Biswas

- Propose a hybrid FWT–AQIM watermarking method optimized for small scale devices
- Utilize YCbCr color space and apply watermark mosaic for redundancy and recovery
- Propose a robust transform-domain features ensuring imperceptibility and resilience
- Achieve real-time performance and strong robustness against various attacks

An Efficient Digital Watermarking Technique for Small Scale devices

Kaushik Talathi[a,*], Aparna Santra Biswas[a,**]

[a]*Department of Computer Science and Engineering , School of Engineering and Technology, COEP Technological University, Pune, Maharashtra, 411005, India*

Abstract

In the age of IoT and mobile platforms, ensuring that content stay authentic whilst avoiding overburdening limited hardware is a key problem. This study introduces hybrid Fast Wavelet Transform & Additive Quantization index Modulation (FWT-AQIM) scheme, a lightweight watermarking approach that secures digital pictures on low-power, memory-constrained small scale devices to achieve a balanced trade-off among robustness, imperceptibility, and computational efficiency. The method embeds watermark in the luminance component of YCbCr color space using low-frequency FWT subbands, minimizing perceptual distortion, using additive QIM for simplicity. Both the extraction and embedding processes run in less than 40 ms and require minimum RAM when tested on a Raspberry Pi 5. Quality assessments on standard and high-resolution images yield PSNR $\geq$ 34 dB and SSIM $\geq$ 0.97, while robustness verification includes various geometric and signal-processing attacks demonstrating near-zero bit-error rates and NCC

*Corresponding author

**Principal corresponding author

Email addresses: kaushiktalathi@gmail.com (Kaushik Talathi), aparna.comp@coeptech.ac.in (Aparna Santra Biswas)

$\geq$ 0.998. Using a mosaic-based watermark, redundancy added enhancing robustness without reducing throughput, which peaks at 11 MP/s. These findings show that FWT-AQIM provides an efficient, scalable solution for real-time, secure watermarking in bandwidth- and power-constrained contexts, opening the way for dependable content protection in developing IoT and multimedia applications.
Keywords: Digital watermarking, FWT, QIM, IoT, Signal processing

1. Introduction

The way we interact with media has been significantly altered by the digital revolution. Without the need for large equipment or conventional studios, people can now create, modify, and share content straight from their mobile devices, courtesy of developments in mobile technology and editing software. Users are now multidimensional participants who are simultaneously makers, consumers, and distributors as a result of this revolution. High-speed internet has further streamlined the sharing and storing of information through social media platforms and streaming services, making it easier for producers to reach large audiences [1, 2]. Digital content can be readily acquired, copied, and distributed illegally through physical transmission channels during communication, data processing, and storage. Copyright protection, content authentication, tamper detection, theft prevention, piracy, and broadcast monitoring and control are just a few of the uses for digital watermarking. By incorporating undetectable data into multimedia assets, including photographs, movies, and music, digital watermarking has become an essential method in resolving these problems by offering authen-

ticity and protection [3, 4]. Furthermore, small-scale devices have become more common, especially in the context of the Internet of Things (IoT), as a result of hardware miniaturization. These little gadgets are now essential in many fields, including home systems, industrial automation, healthcare, and military operations. Because of their extensive use, it is crucial to guarantee the safety and credibility of the digital content they process. Implementing strong security measures is made extremely difficult by these devices' constrained computational capabilities. Lightweight methods that can be easily integrated with limited hardware are crucial to closing this gap [5, 6].

Digital watermarking involves the integration of hidden digital data into the digital content which is known as a watermark [1]. Digital watermarks are often divided into three main classifications: i) Multimedia based, ii) Characteristics based & iii)Application based. Audio, pictures, video, graphics, and text are among the several media carriers in which digital watermarks are placed. Each of these media carriers has distinct signal qualities and attack surfaces that require different embedding techniques. The robustness (robust vs. fragile), visibility (visible vs. invisible), extraction method (blind vs. non-blind), embedding domain (spatial vs. frequency), and adaptivity (adaptive vs. non-adaptive) of watermarks further indicate how well they withstand processing and how noticeable they are within any carrier. These markings are used in broadcast monitoring/control, content authentication, copyright protection, and tamper detection, all of which place unique requirements on watermark fragility, detectability, and strength [7, 8, 6].

To put this concept into practice, each watermarking technique must go through two essential phases: i) Watermark Embedding – In this step, the

watermark is embedded into the host media using a specified algorithm. ii) Watermark Detection/Extraction – In this step, the embedded watermark is checked for detectability and is then extracted from the watermarked media [9].The effectiveness of digital watermarking algorithm/s is evaluated based on three primary criteria: robustness, imperceptibility, and computational efficiency. Robustness refers to the watermark's ability to endure various types of attacks and transformations, while the watermark should be detectable and extraction quality being acceptable. Imperceptibility is another critical criterion, ensuring that the watermark does not degrade the quality of the original content. Computational efficiency is crucial, particularly in resource-constrained environments such as IoT devices or mobile platforms. Addressing this challenge necessitates the development of lightweight, efficient security solutions tailored to the constraints of IoT devices [10, 1].

The highlights of this paper can be summarized as follows:

- Hybrid FWT–AQIM scheme for robust, low-distortion image watermarking.
- Real-time embedding/extraction on Raspberry Pi 5 with less than 40 ms latency.
- Supports image and QR-code watermarks up to 128×128 pixels in host images.
- Evaluates robustness under JPEG, noise, filtering, and geometric attacks.
- Achieves up to 11 MP/s throughput with less than 170 MB memory on IoT devices.

The remainder of this study is organized as follows. The related works are discussed in Section 2. The FWT-AQIM based watermarking for small scale devices is elaborated in Section 3. The experimental results and analysis are presented in Section 4. Finally, Section 5 concludes the study.

2. Related Works

Digital watermarking involves embedding hidden information into digital cover material in a way that allows the data to remain imperceptible yet detectable. The watermark must be resilient against standard signal processing and potential malicious attacks. It serves to uniquely identify the content owner and verify the integrity or authenticity of the carrier signal [10, 11]. Various watermarking systems cater to specific needs: robust watermarks are used for copyright protection; fragile or semi-fragile marks are suited for sensitive fields like medicine, forensics, intelligence, or military; and highly precise embedding is required for content authentication, where even the slightest modification is unacceptable [10].

Data hiding is achieved in both science and the arts via steganography.There are two techniques for embedding media in steganography: i) Spatial Domain and ii) Transform Domain [4]. The digital watermark is directly included into the original signal's pixel values in the spatial domain. Of all spatial domain methods, the Least Significant Bit (LSB) approach is thought to be the most straightforward. The foundation of LSB is watermarking the least significant bits of the original signal [12]. Pixel-based approaches are common in watermarking applications where real-time speed is a primary requirement for their low computational complexity and conceptual simplic-

ity. However, they do have a number of serious shortcomings. Accurate spatial synchronization is necessary to protect users from de-synchronization attacks; neglecting the temporal axis exposes users to multiple frame collusion and video processing; and improving watermarks with only spatial analysis techniques is difficult [13].

The spatial domain representation must first be converted into the frequency domain, and its frequency coefficients must then be adjusted, in order to insert a digital watermark in the frequency domain. Digital watermarking techniques in the transform domain include Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Singular Value Decomposition (SVD) [12]. The watermark is applied throughout the whole original data domain. The host media is translated into the frequency domain using transformation techniques. The altered domain coefficients are then used to hold the watermark data. Lastly, the watermarked media is produced using the inverse transform [13].

Recent advancements in digital watermarking have introduced various Transform domain techniques, such as the DWT and DCT, which effectively address robustness and imperceptibility. However, these methods often involve complex computations, which pose significant challenges for small-scale or resource-constrained devices. Such devices typically have limited processing power, memory, and energy resources, making it difficult to implement computationally intensive watermarking algorithms [13, 14].

The FWT offers a viable alternative by providing similar robustness to DWT and DCT while requiring fewer computational resources. This makes FWT a suitable choice for devices with limited capabilities [11, 12]. Con-

versely, SVD excels in terms of robustness and imperceptibility but demands substantial computational resources, which may not be feasible for small-scale devices [15]. Conventionally, QIM is more efficient in terms of computational demands but tends to compromise on robustness and imperceptibility [16].

To bridge these gaps, integrating different algorithms can yield a more balanced and efficient watermarking solution for small-scale devices. The combination of FWT and QIM is particularly promising as it aims to reconcile computational efficiency with effective performance. This approach addresses the pressing need for practical watermarking solutions that are both robust and efficient, tailored for environments with constrained resources [1].

This study proposes a digital watermarking method that is effective and lightweight, designed for devices with limited resources. The technique strikes a balance between robustness, imperceptibility, and low computing overhead by combining Fast Wavelet Transform (FWT) with Quantization Index Modulation (QIM). This hybrid approach uses the simplicity of QIM and the efficiency of FWT to satisfy the needs of small-scale devices, in contrast to more conventional methods like DWT, DCT, or SVD, which are efficient but resource-intensive. The contribution entails refining the two techniques, creating a hybrid algorithm, and assessing its effectiveness for real-world implementation in IoT and related contexts.

3. Watermarking based on FWT and QIM

The proposed method, which comprises Watermark Embedding and Extraction using a combination of FWT and an AQIM, is explained more thor-

oughly in this section.

3.1. Fast Wavelet transform (FWT)

The FWT is a computationally efficient algorithm for performing multi-resolution signal decomposition. Low-pass and high-pass filtering are applied iteratively over rows and columns, followed by down-sampling, to transform signals from the spatial (or time) domain into the wavelet domain. Consequently, more coarse approximation and detail coefficients enable study at different scales. The original signal is reconstructed from these coefficients using synthesis filters in the inverse transform [17, 18]. As illustrated in Fig. 1, each level captures increasingly coarse image features through multi-level decomposition [19].

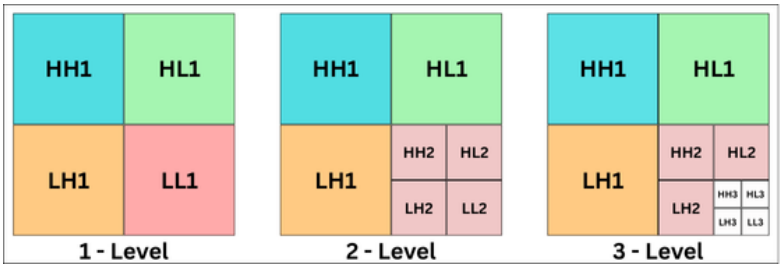


Figure 1: Multiple Levels FWT Decomposition

3.2. Quantization Index Modulation (QIM)

QIM is a well-known watermarking technique that maps host signal coefficients onto one of two interleaved quantization lattices, each of which represents a binary character, in order to encode information. The organized lattice arrangement guarantees resilience against common signal degradations, while the method provides imperceptibility with small perturbations of the original coefficients [16].

During the embedding process, a binary bit chooses one of the two quantization lattices that are defined by step size α. The host coefficient is then snapped to the closest lattice point, and the bit is recovered by testing the (possibly distorted) coefficient for lattice membership later on. The selection of α strikes a balance between robustness (larger α) and imperceptibility (smaller α), which makes QIM both efficient and robust [16, 20].

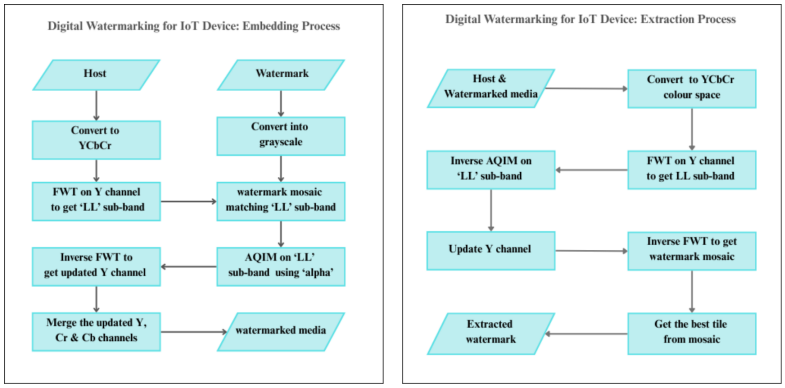
3.3. Watermarking processes

The proposed method supports grayscale watermarks up to 128×128 pixels and is compatible with any host image of size at least $512 \times 512 \times 3$. To adapt to different host–watermark combinations, users can select from multiple wavelet families and decomposition levels and adjust the quantization step α. To minimize perceived color distortion and enable intensity-based processing, the technique transforms RGB images into YCbCr color space, a color representation that divides an image into one luminance component (Y) and two chrominance components (Cb and Cr). This is how the luminance component (Y) is calculated from RGB [7].

$$Luminance(Y) = round(0.299R + 0.587G + 0.114B) \tag{1}$$

The watermark mosaic is constructed in order to precisely match the low-frequency sub-band coefficient dimensions derived from the host image's Y channel's FWT. After that, the mosaic is embedded by changing the corresponding coefficients using AQIM, which maintains robustness and imperceptibility without requiring expensive norm computations by applying a controlled shift. Ultimately, the watermarked image is reconstructed using

an inverse FWT. Fig. 2 demonstrated the embedding process for the proposed method.



Embedding process Extraction process

Figure 2: FWT-AQIM based watermarking process

For extraction, both the original and watermarked images undergo FWT decomposition to retrieve their low-frequency coefficients. Applying the inverse AQIM rule yields a reconstructed watermark mosaic, which is then evaluated using Structural Similarity Index (SSIM) and Normalized Cross-Correlation (NCC). These metrics guide the selection of the best-matching watermark tile, ensuring accurate recovery even under common attacks. Fig. 2 demonstrates the Extraction process for the proposed method.

By combining flexible wavelet settings with AQIM's efficient additive embedding and straightforward subtraction-based extraction, the approach delivers high-quality, resilient watermarks without imposing heavy computational loads.

4. Performance Evaluation

Proposed FWT-AQIM watermarking system implemented using Python on Raspberry Pi 5 having 8 GB of available memory installed with Raspberry Pi OS. For the experiments, a common 64×64 Lena image has been used as a watermark. For the host image, various 512×512 common images like Airplane, Mandril, Peppers, Tiffany, and Cameraman are used as the host image. To widen the reach of experimentation,, an extra QR code as watermark and some non-standard uncommon large size host images are used. This system has to function under strict resource limitations, such as limited CPU, memory, and power, while maintaining data integrity and ownership protection.

4.1. Algorithm Evaluation Metrics

A good evaluation system should take into account the particular performance requirements of small scale devices in addition to more conventional standards like robustness and imperceptibility. The following subsections will highlight various metrics used as part of experimentation to evaluate performance.

4.1.1. Robustness

Robustness is typically evaluated as the the watermark's ability to withstand various attacks using metrics like Bit Error Rate (BER) and Normalized cross-correlation (NCC) of the extracted watermark. A lower BER signifies that the watermark is highly resistant to attacks, whereas NCC values nearing 1 indicate almost perfect similarity.

4.1.2. Imperceptibility

Imperceptibility evaluates the watermark's visual transparency within the host image by comparing the watermarked image and the original host image using Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR). Greater PSNR and SSIM values guarantee that the watermark is practically undetectable to human observers, since they show that the watermarked picture stays very close to the host.

4.1.3. Computational Efficiency

This parameter looks at the watermarking algorithm's efficiency in using system resources for extraction and embedding. Computational efficiency in practice includes several performance metrics:

- Total time: The amount of time needed to carry out extraction or embedding, measured under controlled circumstances to represent actual performance. How the amount of the data affects the execution time.

- Throughput: Usually expressed in megapixels per second (MP/s), throughput is the speed at which data (such as pictures or watermark bits) can be processed. This is essential for preserving system performance on high-demand or real-time systems.

4.2. Base Performance analysis of Algorithm

Table 1 presents a unified evaluation of the algorithm's performance across both standard and non-standard host images of variable sizes, using the Daubechies wavelet (db3) in the Fast Wavelet Transform (FWT) framework with adaptive decomposition levels (Fig. 1) and quantization

steps (α) adjusted based on the watermark type. The algorithm consistently achieves high imperceptibility, with PSNR values ranging from 35 dB to over 40 dB and SSIM scores exceeding 0.99, ensuring near-lossless visual quality. Embedding the Lena watermark at $\alpha = 30$ yields PSNR around 37–38 dB and SSIM > 0.99, while the QR code watermark at $\alpha = 25$ slightly reduces PSNR (34–35 dB) and SSIM (0.97–0.98), but remains visually indistinguishable. Robustness varies with watermark type—QR code watermarking exhibits near-zero BER and NCC $\approx$ 0.998–0.999 across all cases, outperforming Lena watermarking, which typically yields BER around 9–11% and NCC $\approx$ 0.98–0.99. A notable outlier is the Tiffany image with the Lena watermark showing higher BER ($\sim$ 20%) due to specular content, which significantly improves to 4.27% with a QR code watermark. Computational efficiency scales with resolution, with embedding/extraction times from 30 ms to 2 seconds, throughput ranging from 2.8 to 11 MP/s, and memory usage from 157 MB to 330 MB. Despite occasional anomalies (e.g., "Test Image 2" with reduced throughput), the algorithm demonstrates excellent scalability, balancing fidelity, robustness, and resource use, making it well-suited for real-time, resource-constrained environments.

4.3. Simulated Attack Performance

The technique has experimented with a numerous popular simulated attacks, including Gaussian noise, median filter, sandpaper, compression, rotation, scaling, and cropping is covered in this section. Table 2 simulates algorithm performance against these attacks.

4.3.1. Cropping Attack

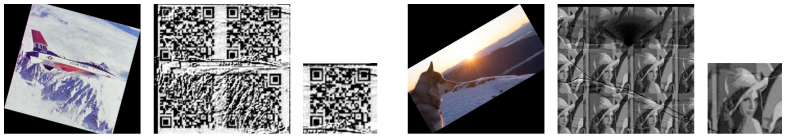
The cropping attack simulation removes a fixed proportion r of pixels from each border and re-centers the remaining $(1-2r) \times (1-2r)$ region on a blank canvas, simulating lost or trimmed image content. For $r = 10\%$, overall 36% of the image area was cropped and centered on a black canvas. Fig. 3 demonstrates the cropping attack on standard and non-standard host images with different crop ratios. BER increases from around 11–15 % to 37 % as the crop ratio r increases from 3% to 20%, while NCC decreases from 0.99 to 0.28 for Lena watermarks.The QR-code watermark shows superior robustness, with BER remaining near 0% and NCC > 0.55 until an extreme crop ratio.

Figure 3: Illustration of the cropping attack

4.3.2. Rotation Attack

The rotation attack turns the image by an angle θ, automatically enlarging the canvas so no corners are lost. Rotation attack rotates the image by angle θ with canvas expansion to avoid clipping of edges. Fig. 4 demonstrates the rotation attack on standard and non-standard host images with different rotation angles. For Lena watermark, small rotations ($\pm 5°$) causes BER rise up to 15%, but only around 4% for QR codes, with NCC higher than 0.9. QR codes preserves BER less than 5% and NCC higher than 0.91 even at

±30° rotations, while Lena watermark observes BER at around 15–17% and NCC around 0.78–0.80.



Airplane-QR: Rotation at 15° Sunrise-Lena: Rotation at -30°

Figure 4: Illustration of the rotation attack

4.3.3. Scaling Attack

The image is first resized by a factor s (up- or down-sampled) and then scaled back to its original dimensions using linear interpolation. This double resampling introduces smoothing, interpolation blur, and anti-aliasing artifacts similar to those seen in repeated resizing operations. Fig. 5 demonstrates the scaling attack on standard and non-standard host images with different scaling factors. When down-scaling to 20%, the NCC falls below 0.5 and the BER increases to about 35% for Lena watermark and 23% for QR watermark. Robustness is preserved by moderate down-scaling to 60% and up-scaling to $\times 1.4$, $\times 2$ for Lena watermark at BER around 27% and NCC higher than 0.74, for QR watermark, BER maintained less than 1% and NCC higher than 0.98.

House-Lena: Scaling at 0.6 Raineir-QR: Scaling at 2

Figure 5: Illustration of the scaling attack

4.3.4. Gaussian Noise Attack

Gaussian Noise attack injects independent, zero-mean Gaussian noise with standard deviation (σ) into each pixel value, modeling sensor imperfections or mild transmission disturbances. If $\sigma = 5$, two-thirds of pixels get shifted by at most ± 5 gray levels, and about 19 out of 20 by at most ± 10 levels. Fig. 6 demonstrates the Gaussian noise attack on standard and non-standard host images with different standard deviations. At $\sigma = 0.5$, image watermarks suffer BER at around 30%, but QR stays at 0% BER & NCC around 0.99. Increasing σ to 10 yields BER at around 32% for Lena watermark and up to 6% for QR watermark, with NCC still higher than 0.89.

House-QR: Gaussian standard deviation at 2 Raineir-Lena: Gaussian standard deviation at 5

Figure 6: Illustration of the gaussian noise attack

4.3.5. JPEG Compression Attack

The image is encoded and then decoded at JPEG quality q. If $q = 30$, file size drops dramatically and visual degradation becomes very apparent. Fig. 7 demonstrates the JPEG compression attack on standard and non-standard host images with different compression values. When compressed lightly (Q = 90), the image watermark experiences about 12% BER while the QR code remains intact at 0% BER; both maintain NCC values exceeding 0.97. Under stronger compression (Q = 30), the QR code still recovers with only about 3–4% BER and NCC above 0.91, whereas the Lena watermark's BER rises to roughly 28% and its NCC falls to about 0.65.

Cameraman-Lena: JPEG compression at 30% Test Image 1-QR: JPEG compression at 70%

Figure 7: Illustration of the JPEG compression attack

4.3.6. Median Filtering Attack

Medial Filter attack applies a $k \times k$ median filter that replaces each pixel with the median value in its neighborhood. This suppresses impulsive noise like salt-and-pepper but also softens edges and fine textures. If k is 5, the filter consists of 25 neighboring pixels for each pixel. Fig. 8 demonstrates the median filter attack on standard and non-standard host images with different medians. A 3x3 median filter retains NCC exceeding 0.95 while producing BER values of about 16–19% for the picture watermark and less than 1% for the QR code. The performance of larger kernels (larger than 7×7) is drastically reduced; for watermark, BER is reaching around 30% and for QR code, BER is reaching about 17%, and NCC can go below 0.78.

Cameraman-QR: Median filter at 5 Test Image 1-Lena: Median filter at 9

Figure 8: Illustration of the median filter attack

4.3.7. Resize Attack

Resize attack downscale image by a factor s, then upscale back to its original size. If s is 5, the image shrinks to 4% of its original area. Almost all

fine details are lost while downscale. Fig. 9 demonstrates the resize attack on standard and non-standard host images with different resize factors. When the image is resized down to 25%, the Lena watermark suffers approximately 27% BER and the QR code about 19%, with NCCs of roughly 0.54 and 0.59, respectively. In contrast, enlarging the image to 150% limits BER to under 19% for the Lena watermark and below 1% for the QR code, while NCC stays above 0.90.

Tiffany-Lena: Resize to 0.25 Maltese-QR: Resize to 1.5

Figure 9: Illustration of the resize attack

4.3.8. Sandpaper Noise (Salt-and-Pepper) Attack

Sandpaper attack randomly sets pixels to black or white with probability p, modeling impulse noise from transmission errors or dust. For p) = 0.01, half a percent of all pixels are set to 0(zero) and the other half percent set to 1(one). Remaining 99% of pixels are kept untouched. Fig. 10 demonstrates the sandpaper attack on standard and non-standard host images with different probabilities. When sandpaper noise is applied at a low density (p = 0.001), the Lena watermark sees about 12–13% BER while the QR code stays below 1% BER, with both retaining NCCs above 0.96. Increasing the noise density to p = 0.05 causes a dramatic drop in performance. The Lena watermark's BER surges to roughly 45%, the QR code's BER rises to about 34%, and NCC falls under 0.46.

Tiffany-QR: Sandpaper p=0.001 Maltese-Lena: Sandpaper p=0.01

Figure 10: Illustration of the sandpaper attack

Table 1: Algorithm simulation on standard (512×512, level = 2, wavelet = db3) & non-standard (variable size and parameters) Host images.

	Imperceptibility		**Robustness**		**Embedding**			**Extraction**		
Host Image & parameter details	PSNR (dB)	SSIM	BER (%)	NCC	Time (s)	Memory (MB)	TP (MP/s)	Time (s)	Memory (MB)	TP (MP/s)
Airplane (Lena-WM), $\alpha = 30$	37.30	0.9969	9.79	0.9904	0.031	157	8.27	0.032	163	8.27
Airplane (QR-WM), α=25	34.53	0.9698	0.00	0.9984	0.040	157	6.44	0.034	163	6.44
House (Lena-WM), $\alpha = 30$	37.31	0.9976	9.79	0.9904	0.039	157	8.27	0.032	158	6.62
House (QR-WM), $\alpha = 25$	34.54	0.9777	0.00	0.9984	0.028	157	9.04	0.033	163	9.04
Peppers (Lena-WM), $\alpha = 30$	37.35	0.9962	9.79	0.9962	0.032	158	8.13	0.036	164	7.34
Peppers (QR-WM), $\alpha = 25$	34.56	0.9749	0.00	0.9983	0.037	158	7.01	0.034	159	7.71
Tiffany (Lena-WM), $\alpha = 30$	38.28	0.9965	20.31	0.8713	0.032	158	8.08	0.034	164	7.78
Tiffany (QR-WM), $\alpha = 25$	35.58	0.9773	4.27	0.9387	0.034	158	7.73	0.035	164	7.57
Cameraman (Lena-WM), $\alpha = 30$	37.43	0.9914	9.79	0.9904	0.031	157	8.24	0.033	163	8.24
Cameraman (QR-WM), $\alpha = 25$	34.60	0.9577	0.00	0.9984	0.032	157	7.99	0.033	158	7.99
Maltese (Lena-WM, 2446×2238) ($\alpha = 90$, Level = 4, Wavelet db3)	40.39	0.9993	11.52	0.9801	0.60	295	9.18	0.47	296	11.62
Maltese (QR-WM, 2446×2238) ($\alpha = 90$, Level = 4, Wavelet db3)	38.12	0.9964	0.00	0.9989	0.60	294	9.07	0.49	295	11.21
Rainier (Lena-WM, 1080×1920) ($\alpha = 60$, Level = 3, Wavelet db3)	37.4	0.9960	10.62	0.9903	0.23	206	8.99	0.202	207	10.26
Rainier (QR-WM, 1080×1920) ($\alpha = 35$, Level = 3, Wavelet db3)	34.99	0.9905	0.00	0.9987	0.231	206	8.97	0.207	254	10.01
Sunrise (Lena-WM, 2908×6000) ($\alpha = 90$, Level = 4, Wavelet db3)	40.59	0.9986	11.52	0.9801	2.195	252	7.95	1.916	330	9.11
Sunrise (QR-WM, 1080×1920) ($\alpha = 70$, Level = 4, Wavelet db3)	38.54	0.9960	0.00	0.9989	2.175	252	8.02	1.900	330	9.18
Test Image 1 (Lena-WM, 4390×2926) ($\alpha = 90$, Level = 4, Wavelet db3)	40.32	0.9988	11.52	0.9801	1.457	225	8.82	1.341	251	9.57
Test Image 1 (QR-WM, 4390×2926) ($\alpha = 70$, Level = 4, Wavelet db3)	38.18	0.9955	4.27	0.9989	1.434	225	8.96	1.384	250	9.28

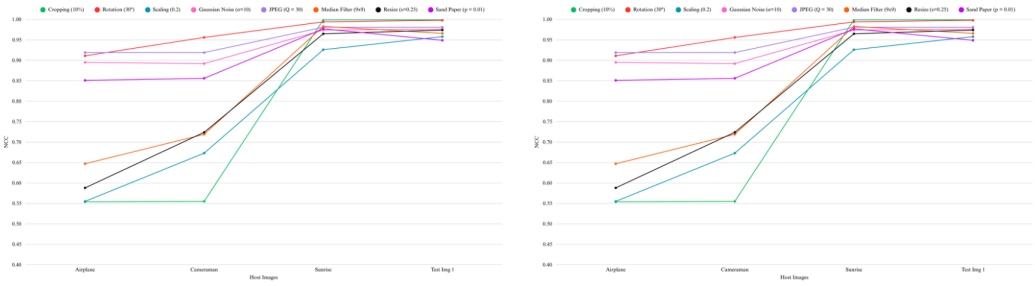
Across all experiments—from baseline embedding tests (Table 1) to simulated attack scenarios (Table 2)—the proposed FWT–AQIM watermarking scheme consistently demonstrates a well-balanced trifecta of imperceptibility, robustness, and computational efficiency across both standard and non-standard host images. Utilizing the Daubechies wavelet (db3) with adaptive decomposition levels and embedding in the luminance (Y) channel of the YCbCr color space, the method ensures high energy compaction and visual quality preservation. Embedding a 64×64 Lena watermark yields PSNR val-

Table 2: Algorithm performance under various attacks on standard and non standard images

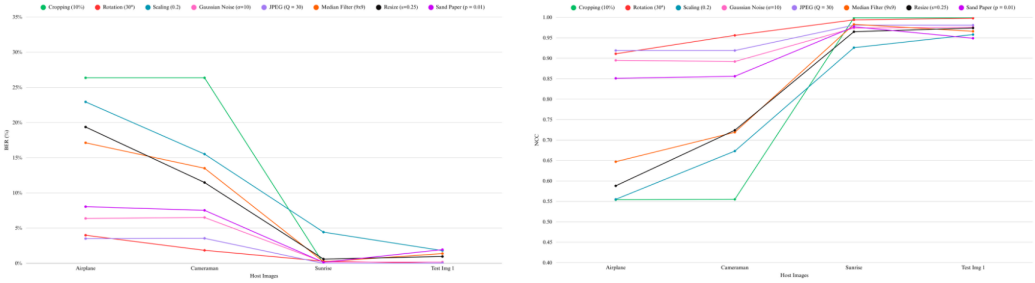
Attack Details	Airplane				Cameraman				Sunrise				Test Img 1			
	Lena		QR		Lena		QR		Lena		QR		Lena		QR	
	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC	BER	NCC
Cropping (10%)	26.78	0.379	26.37	0.554	26.73	0.380	26.37	0.555	11.52	0.980	0.00	0.999	11.52	0.980	0.00	0.999
Cropping (20%)	36.89	0.279	42.06	0.403	36.79	0.280	42.04	0.405	12.45	0.974	0.00	0.995	11.52	0.980	0.00	0.999
Rotation (30°)	15.50	0.795	4.00	0.911	15.26	0.784	1.85	0.956	12.59	0.953	0.31	0.994	11.49	0.978	0.00	0.998
Rotation (-30°)	16.53	0.762	4.13	0.910	15.38	0.781	1.88	0.956	12.77	0.953	0.29	0.989	11.79	0.978	0.85	0.976
Scaling (s=0.2)	34.03	0.405	22.95	0.555	30.59	0.490	15.53	0.673	32.45	0.831	4.44	0.926	23.71	0.795	1.81	0.958
Scaling (s=0.6)	24.39	0.733	4.10	0.923	24.93	0.756	3.10	0.937	28.95	0.969	0.00	0.996	22.56	0.967	0.00	0.996
Scaling (s=2)	12.28	0.915	0.31	0.987	12.35	0.900	0.61	0.984	10.45	0.979	0.00	0.999	10.38	0.977	0.00	0.999
Gauss (σ=0.5)	31.47	0.985	0.00	0.997	31.54	0.986	0.00	0.997	38.31	0.970	0.00	0.996	37.31	0.974	0.00	0.997
Gauss (σ=5.0)	27.22	0.829	0.59	0.966	26.95	0.838	0.59	0.966	36.23	0.954	0.00	0.991	34.91	0.952	0.00	0.991
Gauss (σ=10)	32.08	0.321	6.39	0.895	31.27	0.603	6.52	0.892	33.30	0.889	0.19	0.974	32.22	0.897	0.19	0.976
JPEG (Q=70)	16.58	0.905	0.00	0.987	17.97	0.891	0.00	0.986	15.45	0.946	0.00	0.995	14.14	0.939	0.00	0.995
JPEG (Q=30)	27.66	0.654	3.52	0.919	26.81	0.689	3.56	0.919	20.02	0.879	0.00	0.981	16.09	0.876	0.04	0.981
Median (5x5)	20.89	0.658	3.86	0.895	20.43	0.759	3.19	0.910	17.53	0.957	0.00	0.989	15.99	0.943	0.02	0.995
Median (9x9)	29.52	0.478	17.14	0.647	27.69	0.538	13.52	0.719	17.16	0.928	0.24	0.983	16.26	0.828	1.39	0.966
Resize (S=0.25)	31.23	0.439	19.38	0.588	27.29	0.536	11.49	0.724	25.98	0.869	0.61	0.965	27.56	0.874	0.98	0.974
Resize (S=0.75)	21.17	0.841	2.32	0.962	20.19	0.872	1.90	0.965	26.39	0.972	0.00	0.997	25.44	0.967	0.00	0.996
Resize (S=1.5)	18.55	0.915	0.49	0.985	16.69	0.903	0.81	0.983	22.46	0.979	0.00	0.999	15.79	0.974	0.00	0.998
Sandpaper (P=0.005)	21.17	0.726	4.29	0.919	20.19	0.702	3.73	0.929	16.55	0.969	0.00	0.989	16.16	0.868	0.56	0.978
Sandpaper (P=0.01)	25.85	0.562	8.06	0.851	25.27	0.569	7.54	0.856	15.69	0.836	0.15	0.977	18.97	0.772	1.95	0.949
Sandpaper (P=0.05)	45.04	0.155	33.59	0.455	40.79	0.239	26.44	0.511	38.94	0.296	18.09	0.651	30.96	0.433	22.48	0.567

Table 3: Robustness comparison between proposed method and existing methods based on NCC values

Attacks	Proposed Method	Sk A, et al.[2]	Kumar Shrivastava S, et al.[21]	Li Z, et al. [22]
Cropping	0.999	0.9973	0.9869	0.9820
Rotation	0.997	-	0.8562	0.9682
Scaling	0.993	-	0.7212	1.0000
Gaussian Noise	0.997	0.7114	0.7523	0.9676
JPEG Compression	0.999	-	0.7823	1.0000
Median Filter	0.992	0.8518	-	0.9908
Sand paper	0.995	0.9485	-	0.9350



BER & NCC values with watermark Lena



BER & NCC values with watermark QR code

Figure 11: Graphical representation of various attacks for standard and non-standard images

ues around 37–38 dB, SSIM > 0.99, BER around 9–12%, and NCC ≈ 0.99, while QR code watermarking offers slightly lower PSNR (34–35 dB) and SSIM (0.97–0.98), but achieves near-zero BER and NCC up to 0.998—all under 40 ms, within 160 MB memory, and at throughput ranging from 7–11 MP/s. Under attack conditions, QR code watermarks maintain superior robustness (BER $< 5\%$, NCC > 0.90), even under severe distortions, while Lena watermarks degrade more noticeably (BER up to 42%, NCC ≈ 0.75). Similar resilience is noted across high-resolution, non-standard hosts where deeper FWT levels and more mosaic tiles enhance embedding redundancy. QR codes remain highly robust (BER near zero, NCC > 0.94), whereas

Table 4: Watermark embedding and extraction time (in seconds) for the proposed method and existing method for IoT devices

Standard images	**Proposed Method**		**Sk A, et al.[2]**	
	Embedding	**Extraction**	**Embedding**	**Extraction**
Peppers	0.032	0.036	0.056892	0.029547
Baboon	0.033	0.034	0.020165	0.034652
Aeroplane	0.031	0.032	0.048915	0.025395

Lena BER increases to ∼20% under moderate distortion. Noise resilience is also favorable: at low sandpaper noise (p = 0.001), Lena BER is ∼12% vs. <1% for QR; under extreme noise (p = 0.05), both degrade, but QR remains more robust. Fig. 11 illustrates these trends, showing how larger hosts support higher FWT levels and improved embedding granularity, especially benefitting structured watermarks like QR codes. Timing results illustrated in Table 4 indicate our method completes embedding/extraction 0.03–0.04s faster than Sk A et al. [2], while maintaining high NCC values ($\geq$ 0.992) across all seven attack types (Table 3), including near-perfect recovery under scaling and JPEG compression. Altogether, FWT–AQIM offers a scalable, lightweight, and resilient watermarking solution, ideal for real-time, resource-constrained applications such as IoT systems.

5. Conclusion and Future scope

In this study, the FWT–AQIM watermarking scheme maintains an ideal balance of speed, strength, and invisibility—exactly what is needed for de-

vices with limited memory and power. The watermark is hidden in the low-frequency layer of the luminance channel (Y) in YCbCr space, allowing embedding and extraction of data in less than 40 ms on a Raspberry Pi 5 while minimizing visual distortion (PSNR $\geq$ 34 dB, SSIM $\geq$ 0.97). The mosaic-based spread that exploit spatial redundancy, QR-code watermarks appear nearly faultless (NCC $\geq$ 0.998) even when images undergo significant cropping, rotation, scaling, compression, noise, or filtering. With a peak throughput of 11 MP/s across a range of image sizes, it surpasses similar techniques without compromising fidelity. These findings demonstrate that FWT-AQIM based method is not only workable in low-power, real-time settings but also resilient enough to safe-guard digital content.

In the future, the system might be expanded to include adaptive quantization parameter adjusting based on local texture complexity may enhance the transparency and resilience of watermarks even further. It may be possible to further minimize localized distortions and enhance watermark clarity under specific degradations by using mosaic-based watermark reconstruction, in which all eligible tiles over a predetermined threshold are superimposed to create a single representative tile.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Kaushik Talathi: Conceptualization, Methodology, Formal analysis, Data curation, Software, Validation, Visualization, Writing – original draft. **Aparna Santra Biswas**: Conceptualization, Methodology, Formal analysis, Visualization, Investigation, Writing – review & editing, Supervision.

Data availability

The data will be available as on request.

References

[1] Begum M, Uddin MS. *Digital Image Watermarking Techniques: A Review*. Information 2020;11:110. https://doi.org/10.3390/info11020110.

[2] Sk A, Masilamani V. *A novel digital watermarking scheme for data authentication and copyright protection in 5G networks*. Computers & Electrical Engineering 2018;72:589–605. https://doi.org/10.1016/j.compeleceng.2018.02.045.

[3] Kumar A. *A Review on Implementation of Digital Image Watermarking Techniques Using LSB and DWT*. In: Tuba M, Akashe S, Joshi A, editors. Information and Communication Technology for Sustainable Development, vol. 933, Singapore: Springer Singapore; 2020, p. 595–602. https://doi.org/10.1007/978-981-13-7166-0_59.

[4] Al-Kadei FHMS, Hasan SN. *Improve a secure blind watermarking technique for digital video*. PEN 2022;10:283. https://doi.org/10.21533/pen.v10i2.2859.

[5] Kaw JA, Loan NA, Parah SA, et al. *A reversible and secure patient information hiding system for IoT driven e-health*. International Journal of Information Management 2019;45:262–75. https://doi.org/10.1016/j.ijinfomgt.2018.09.008.

[6] Reyes-Ruiz L, Fragoso-Navarro D, Garcia-Ugalde F, et al. *Robust Dual Digital Watermark Applied to Antique Digitized Cinema Images: Resistant to Print-Scan Attack*, JOIG 2023;11:61–71. https://doi.org/10.18178/joig.11.1.61-71.

[7] Yao Y, Zhang W, Wang H, et al. *Content-adaptive reversible visible watermarking in encrypted images*. Signal Processing 2019;164:386–401. https://doi.org/10.1016/j.sigpro.2019.06.034.

[8] Xie X, Jiang J, Zhang J, et al. *Reversible adversarial visible image watermarking*. Signal Processing 2025;234:109999. https://doi.org/10.1016/j.sigpro.2025.109999.

[9] Kumar S, Singh BK, Yadav M. *A Recent Survey on Multimedia and Database Watermarking*. Multimed Tools Appl 2020;79:20149–97. https://doi.org/10.1007/s11042-020-08881-y.

[10] Kapse AS, Belokar S, Gorde Y, et al. *Digital Image Security Using Digital Watermarking* IRJET 2018:5(3):163–166.

[11] Gupta G, Gupta VK, Chandra M. *An efficient video watermarking based security model*. Microsyst Technol 2018;24:2539–48. https://doi.org/10.1007/s00542-017-3689-x.

[12] Fkirin A, Attiya G, El-Sayed A, et al. *Copyright protection of deep neural network models using digital watermarking: a comparative study*. Multimed Tools Appl 2022;81:15961–75. https://doi.org/10.1007/s11042-022-12566-z.

[13] Hussein Tuama Hazim, Nawar Alseelawi, ALRikabi HTHS. *A Novel Method of Invisible Video Watermarking Based on Index Mapping and Hybrid DWT-DCT*. Int J Onl Eng 2023;19:155–73. https://doi.org/10.3991/ijoe.v19i04.37581.

[14] Panyavaraporn J, Horkaew P. *DWT/DCT-based Invisible Digital Watermarking Scheme for Video Stream*. 2018 10th International Conference on Knowledge and Smart Technology (KST), Chiang Mai: IEEE; 2018, p. 154–7. https://doi.org/10.1109/KST.2018.8426150.

[15] Kothari AM, Dwivedi V, Thanki RM. *Singular Value Decomposition (SVD)-Based Video Watermarking*. Watermarking Techniques for Copyright Protection of Videos. Cham: Springer International Publishing; 2019, p. 63–80. https://doi.org/10.1007/978-3-319-92837-1_4.

[16] Zareian M, Tohidypour HR. *Robust quantisation index modulation-based approach for image watermarking*. IET Image Processing 2013;7:432–41. https://doi.org/10.1049/iet-ipr.2013.0048.

[17] Gomes J, Velho L. *The Fast Wavelet Transform*. From Fourier Analysis to Wavelets, Cham: Springer International Publishing; 2015, p. 89–100. https://doi.org/10.1007/978-3-319-22075-8_7.

[18] Arya AS, Saha A, Mukhopadhyay S. *ADMM optimizer for integrating wavelet-patch and group-based sparse representation for image inpainting*. Vis Comput 2024;40:345–72. https://doi.org/10.1007/s00371-023-02786-1.

[19] Vaidya SP, Kishore VR. *Adaptive Medical Image Watermarking System For E-Health Care Applications*. SN COMPUT SCI 2022;3:107. https://doi.org/10.1007/s42979-021-00995-w.

[20] [1] Kim D-W, Kim J-K, Piao Z, et al. *Transient detection-based adaptive audio watermarking using attack-aware optimization*. Digital Signal Processing 2024;146:104352. https://doi.org/10.1016/j.dsp.2023.104352.

[21] Kumar Shrivastava S, S.K. Mahendran D. *Digital Watermarking using Lifting Wavelet Transform of Crowd Sourced Images* IJET 2018;7:357. https://doi.org/10.14419/ijet.v7i2.20.16733.

[22] Li Z, Zhang H, Liu X, et al. *Blind and safety-enhanced dual watermarking algorithm with chaotic system encryption based on RHFM and DWT-DCT* Digital Signal Processing 2021;115:103062. https://doi.org/10.1016/j.dsp.2021.103062.