Navaneedha Rishnan K, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science,
Engineering and Technology

An Open Access Journal

Harnessing Convolutional Neural Networks for Robust Digital Image Watermarking

Navaneedha Rishnan K[1], Professor DR.S.Nagasundaram[2]
Department of computer Application
Vels Institute of Science, technology & Advanced Studies
VISTAS Pallavaram, Chennai, Tamilnadu, India

Abstract- In the era of digital communication and multimedia sharing, ensuring the integrity and ownership of digital content has become increasingly crucial. Digital watermarking techniques offer a solution by embedding imperceptible yet detectable signals within multimedia content, serving as a form of copyright protection and authentication. This research presents a novel approach to digital watermarking using convolutional neural networks (CNNs). The proposed technique involves the training of a CNN model to embed binary watermarks into images, followed by a demodulation and extraction process to recover the watermark from watermarked images. Evaluation metrics such as Bit Error Rate (BER), Mean Squared Error (MSE), and Peak Signal-to-Noise Ratio (PSNR) are employed to assess the fidelity of the watermarked images and the accuracy of watermark extraction. Comparative analysis with traditional watermarking techniques demonstrates the effectiveness of the CNN-based approach in terms of robustness and imperceptibility. The results showcase the potential of CNNs in enhancing the security and authenticity of digital content through watermarking, paving the way for advanced applications in digital rights management and content authentication.

Keywords— DWT, convolutional neural network, Robustness Evaluation, deep learning, Content Authentication.

I. INTRODUCTION

Most multimedia signals today are in digital formats which are easy to reproduce and modify without leaving any trace of manipulations. It is therefore very simple to tamper with any image and make it available to others. Authentication technologies fulfill an increasing need for trustworthy digital data in commerce, industry and defense. Watermarking has become a popular technique for copyright enforcement and image authentication.

Here, an effort has been made to present a novel method for image authenticationwith localization for the purpose of tamper detection.

ELEMENTS OF A WATERMARKING SYSTEM

According to a widespread point of view, a watermarking system is much like a communication system consisting of three main elements: a transmitter, a communication channel, and a receiver [1]. To be more specific, the embedding of the to-be-hidden information within the host signal plays the role of data transmission; any processing applied to the host data after information concealment, along with the interaction between the concealed data and the host data itself, represents the transmission through a communication channel; the recovery of the hidden information from the host data acts the part of the receiver. By following the communication analogy, any watermarking system assumes the form given in Fig.1.1.

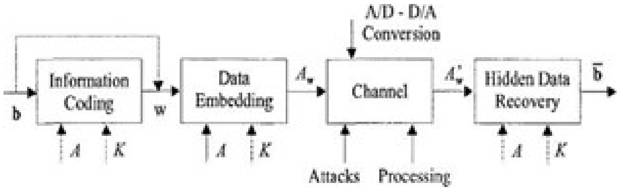


Fig.1 Data Hiding System

With bk taking values in {0, 1}. The string b is referred to as the watermark code. At the transmitter side, a data embedding module inserts the string b within a piece of data called host data or host signal. The host signal may be of any media type: an audio file, a still image, and a piece of video or a combination of the above. The embedding module may accept a secret key K as an additional input. Such a key, whose main goal is to introduce some secrecy within the embedding step, is usually used to parameterize the embedding process and make the recovery of the watermark impossible for unauthorized users which do not have access to K. The functionalities of the data embedding module can be further split into three main tasks: (i) information coding; (ii) watermark embedding; (iii) watermark concealment.

INFORMATION CODING

In many watermarking system, the information message b is not embedded directly within the host signal. On the contrary, before insertion, vector b is transformed into awatermark signal w = {w1,,w2 ... wn} which is more suitable for embedding [2].

In a way that closely resembles a digital communication system, the watermark code b may be used to modulate a much longer spread-spectrum sequence, it may be transformed into a bipolar signal where zero's are mapped in +1 and one's in —1, or it may be mapped into the relative position of two or more pseudo- random signals in the case of position-encoded- watermarking [3].
Eventually, b may be left as it is, thus leading to a scheme in which the watermark code is directly inserted within A, the host image. In this case, the watermark signal w coincides with the watermark code b. Before transforming the watermark code into the watermark signal, b may be channel-coded to increase robustness against possible attacks. As a matter of fact, it turns out that channel coding greatly improves the performance of any watermarking system.

WATERMARK EMBEDDING

In watermark embedding, or watermark casting, an embedding function e takes the host asset A, the watermark signal w, and, possibly, a key K, and generates the watermarked asset Aw:

$$\varepsilon(A, \mathbf{w}, K) = A_{\mathbf{w}} \qquad (1.2)$$

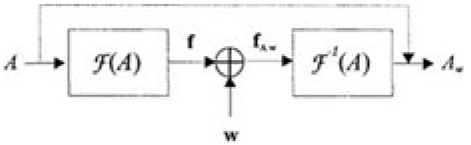


Fig.2 Watermark embedding via invertible feature extraction.

–It is to be noted that the above equation still holds when the watermark code isembedded directly within A, since in this case w = b. The definition of e usually goes through the selection of a set of asset features, called host features, which are modified according to the watermark signal. By letting the host features be denoted by(A) = fA = {f1, f2
... fm} watermark embedding amounts to the definition of an insertion operator Å which transforms ¦ (A) into the set of watermarked features

$$f(A_{w}) = f(\,\varepsilon(A, \mathbf{w}, K)) = f(A) \oplus \mathbf{w} \;\ldots\ldots\ldots (1.3)$$

In general m ^{1} n, i.e. the cardinality of the host feature set need not be equal to the watermark signal length.

Though equations (1.2) and (1.3) basically describe the same process, namely watermark casting within A, they tend to view the embedding problem from two different perspectives. According to equation (1.2), embedding is more naturally achieved by operating on the host asset, i.e. e modifies A so that when the feature extraction function ¦ is applied to Aw, the desired set of features fAw = {fw1,
¦w2...¦wm} is obtained.
Equation (1.3) describes the watermarking process as a direct modification of fA through the embedding operator Å. According to this formulation, the watermark embedding process assumes the form shown in Fig.1.2. First the host feature set is extracted from A, then the Å operator is applied producing

fAw, finally the extraction procedure is inverted to obtain Aw:

$$A_w = F^{-1}(f_{A_w}) \qquad (1.4)$$

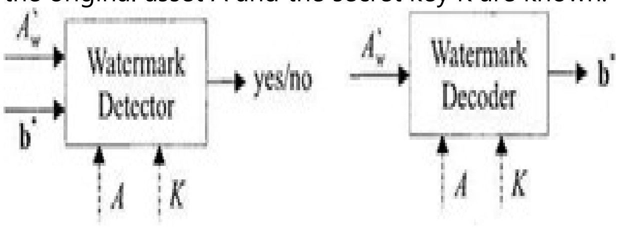
WATERMARK CONCEALMENT

The main concern of the embedding part of any data hiding system is to make the hidden data imperceptible. This task can be achieved either implicitly, by properly choosing the set of host features and the embedding rule, or explicitly, by introducing a concealment step after watermark embedding [4]. To this aim, the properties of the human senses must be carefully studied, since imperceptibility ultimately relies on the imperfections of such senses. Thereby, still image and video watermarking rely on the characteristics of the Human Visual System (HVS).

RECOVERY OF THE HIDDEN INFORMATION

The receiver part of the watermarking system may assume two different forms. According to the scheme reported in Fig.1.3 (a), the watermark detector reads A'w and a watermark code b*, and decides whether A'w contains b* or not. The detector may require that the secret key K used to embed the watermark is known [5]. In addition,the detector may perform its task by comparing the watermarked asset A'w with the original, non-marked, asset A, or it may not need to know A to take its decision. In the latter case, it is said that the detector is blind, whereas in the former case the detectoris said to be non-blind.

Alternatively, the receiver may work as in Fig.1.3 (b). In this case the aim of the receiver is to extract b* from A'w and the watermark code b* is not known in advance As before, the extraction may require that the original asset A and the secret key K are known.



(a) Detectable watermarking (b) Readable watermarking Fig.3: Watermark Recovery.

II. EXISTING METHODS

HAAR WAVELET BASED SYSTEM

The watermarking model described implements a robust and efficient technique for embedding and extracting watermarks within digital images. Initially, the model loads and resizes both the original color image and the watermark image, ensuring compatibility in dimensions. Employing the Discrete Wavelet Transform (DWT) with the Haar wavelet, the model decomposes both images into four sub-bands, facilitating efficient data representation. During watermarking, the model modifies the low-frequency LL sub-band of the host image by adding a scaled version of the watermark's LL sub-band. This process seamlessly integrates the watermark into the host image while preserving its visual integrity.

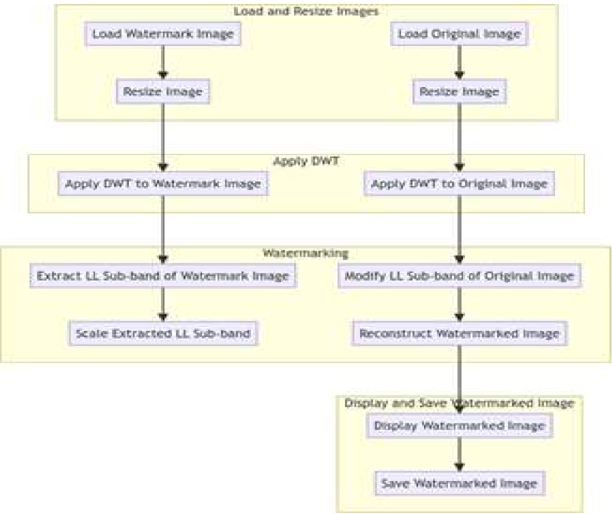

Load and Resize Images
Load Watermark Image
Load Original Image
Resize Image
Resize Image
Apply DWT
Apply DWT to Watermark Image
Apply DWT to Original Image
Watermarking
Extract LL Sub-band of Watermark Image
Modify LL Sub-band of Original Image
Scale Extracted LL Sub-band
Reconstruct Watermarked Image
Display and Save Watermarked Image
Display Watermarked Image
Save Watermarked Image

Figure.4 Watermark Embedding Process.

Subsequently, watermark extraction involves subtracting the LL sub-band of the original image from that of the watermarked image, followed by scaling to isolate the embedded watermark. Evaluation metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) ensure the fidelity of the extracted watermark. Furthermore, post-processing steps, including conversion to grayscale and median filtering, enhance the clarity and robustness of the extracted watermark. This watermarking model presents a comprehensive approach to digital watermarking, offering both embedding and extraction functionalities with considerations for accuracy and resilience against common image processing operations. Figure shows each step in the watermark embedding process outlined in the flowchart:

1. Load and Resize Images:

In this initial step, the original image intended for watermark embedding is loaded into memory. This image is then resized to a desired dimension to ensure compatibility with the watermark image. Additionally, the watermark image, which contains the desired information to be embedded, is also loaded and resized to match the dimensions of the original image. This ensures that the watermark can be seamlessly integrated into the host image without distortion.

2. Apply DWT (Discrete Wavelet Transform):

Utilizing the Discrete Wavelet Transform (DWT), the original image undergoes a multi-resolution analysis, resulting in the decomposition of the image into its constituent frequency bands. This process generates four sub- bands: LL (low-low), LH (low-high), HL (high-low), and HH (high-high). Similarly, the watermark image is subjected to the DWT, allowing it to be represented in a frequency domain suitable for embedding.

3. Watermarking:

The watermark embedding process begins by modifying the LL sub-band of the original image. This modification involves adding a scaled version of the LL sub-band of the watermark image to the LL sub-band of the original image. By doing so, the information encoded within the watermark is introduced into the host image while preserving its visual integrity. This step is crucial for ensuring that the watermark is robustly embedded and remains perceptually invisible within the host image.

4. Reconstruct Watermarked Image:

After embedding the watermark into the original image, the watermarked image is reconstructed using inverse Discrete Wavelet Transform (IDWT). This process combines the modified LL sub-band of the original image with the original LH, HL, and HH sub-bands to produce the final watermarked image. The reconstructed image retains the visual characteristics of the original image while incorporating the embedded watermark, ready for further processing or distribution.

5. Display and Save Watermarked Image:

The watermarked image is displayed to visualize the result of the watermark embedding process. Additionally, the watermarked image is saved to a file for future reference or distribution. This step ensures that the embedded watermark is effectively integrated into the host image, ready for subsequent extraction or analysis.

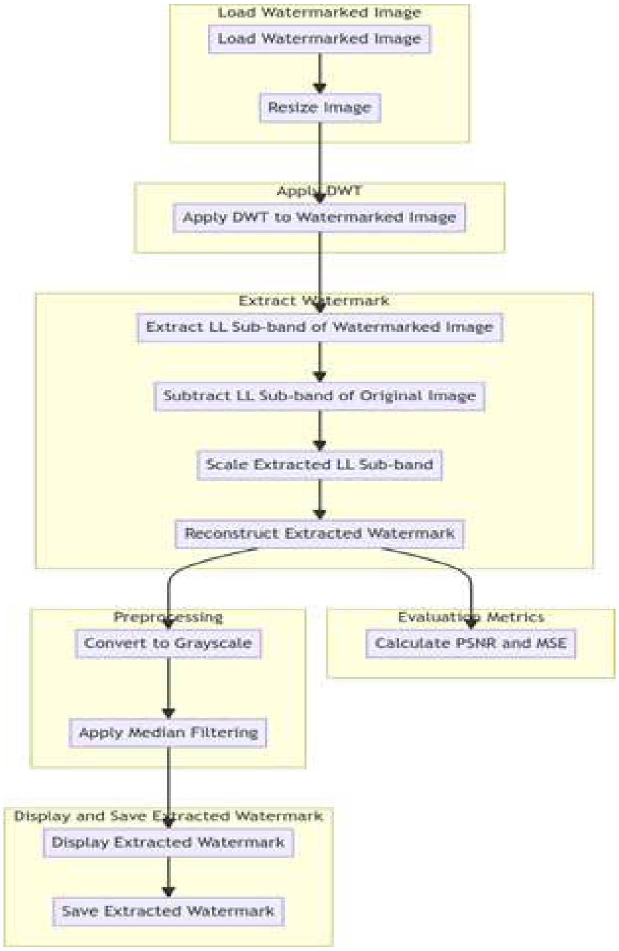

Load Watermarked Image
Load Watermarked Image
Resize Image
Apply DWT
Apply DWT to Watermarked Image
Extract Watermark
Extract LL Sub-band of Watermarked Image
Subtract LL Sub-band of Original Image
Scale Extracted LL Sub-band
Reconstruct Extracted Watermark
Preprocessing
Convert to Grayscale
Apply Median Filtering
Evaluation Metrics
Calculate PSNR and MSE
Display and Save Extracted Watermark
Display Extracted Watermark
Save Extracted Watermark

Figure.5 Watermark Extraction Process.

1.Load Watermarked Image:

The watermark extraction process begins by loading the watermarked image, which contains the embedded watermark. This image is then resized if necessary to ensure consistency with the processing requirements.

2. Apply DWT (Discrete Wavelet Transform):

The loaded watermarked image undergoes the Discrete Wavelet Transform (DWT) to decompose it into its constituent frequency bands. This results in the extraction of the LL (low-low), LH (low-high), HL (high-low), and HH (high-high) sub-bands, which are essential for recovering the embedded watermark.

3. Extract Watermark:

The extraction of the watermark involves isolating the LL sub-band of the watermarked image, which contains the embedded watermark information. This sub-band is then processed by subtracting the LL sub-band of the original image, which serves to remove the original content and isolate the embedded watermark. Subsequently, the extracted LL sub-band is scaled to recover the original watermark information.

4. Evaluation Metrics:

To assess the fidelity of the extracted watermark, evaluation metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) are calculated. These metrics provide quantitative measures of the similarity between the original watermark and the extracted watermark, aiding in evaluating the effectiveness of the extraction process.

5. Preprocessing:

In preparation for visualization and further analysis, the extracted watermark undergoes preprocessing steps. Firstly, it is converted to grayscale to simplify its representation. Additionally, median filtering is applied to the grayscale watermark to enhance its clarity and reduce noise, ensuring that the extracted watermark is of high quality and suitable for further processing.

6. Display and Save Extracted Watermark:

Finally, the extracted watermark is displayed to visualize the result of the extraction process. Additionally, the extracted watermark is saved to a file for documentation or further analysis. This step ensures that the embedded watermark can be successfully recovered from the watermarked image, validating the effectiveness of the watermarking and extraction processes.

In summary, the watermarking process begins with the loading and resizing of both the original image and the watermark image to ensure compatibility. These images then undergo the Discrete Wavelet Transform (DWT), decomposing them into frequency bands necessary for embedding the watermark. The watermark is embedded into the original image by modifying its LL sub-band with a scaled version of the watermark's LL sub-band. After reconstruction, the watermarked image is displayed and saved for further use. In the extraction process, the watermarked image is loaded and subjected to DWT to extract the embedded watermark from its LL sub-band. Evaluation metrics such as PSNR and MSE are calculated to assess the fidelity of the extracted watermark. Preprocessing steps, including conversion to grayscale and median filtering, are applied to enhance the quality of the extracted watermark. Finally, the extracted watermark is displayed and saved for analysis. This comprehensive process enables the robust embedding and extraction of watermarks in digital images, facilitating tasks such as copyright protection and content authentication.

III. PROPOSED WORK

In the rapidly expanding digital landscape, the protection of intellectual property rights and the prevention of content piracy have become paramount concerns. Traditional watermarking techniques, while effective to a degree, often struggle to cope with the scale and diversity of digital media. To address these challenges, this paper proposes a novel approach leveraging Convolutional Neural Networks (CNNs) for automated noise modulation in image watermarking. CNNs have demonstrated remarkable success in various computer vision tasks, offering the potential to automate and optimize the watermarking process while enhancing robustness and efficiency. This paper aims to develop a comprehensive methodology for CNN-based image watermarking, encompassing data preparation, CNN architecture design, training, testing, and evaluation. Each step in the methodology is carefully designed and executed to develop a robust and effective CNN-based image watermarking technique.

Watermark embedding is a fundamental process in digital watermarking, whereby a unique identifier or signal, known as the watermark, is invisibly embedded into the host multimedia content to assert ownership, authenticate the content, or convey additional information. This process is crucial for protecting intellectual property rights, combating content piracy, and ensuring the integrity and authenticity of digital assets.

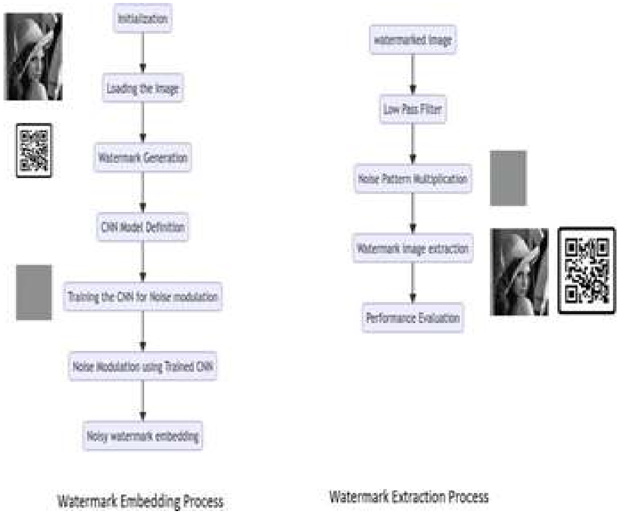

Initialization
Loading the Image
Watermark Generation
CNN Model Definition
Training the CNN for Noise modulation
Noise Modulation using Trained CNN
Noisy watermark embedding
watermarked Image
Low Pass Filter
Noise Pattern Multiplication
Watermark image extraction
Performance Evaluation
Watermark Embedding Process
Watermark Extraction Process

Figure.6 : Proposed Process Flow of the Methodology.

The watermark embedding process involves several key steps to effectively conceal the watermark within the host multimedia content while minimizing perceptible distortion. First, the host content, such as images, videos, or audio files, is preprocessed to prepare it for watermark embedding.

This preprocessing step may involve normalization, resizing, or other transformations to ensure consistency and compatibility with the watermark embedding algorithm. Next, the watermark signal is modulated onto the host content using a specified embedding algorithm or technique.

This process involves modifying certain features or characteristics of the host content to embed the watermark signal while minimizing perceptual impact. The embedding algorithm typically operates in the spatial or transform domain, altering pixel values, frequency coefficients, or other relevant parameters to encode the watermark information.

Once the watermark signal is embedded into the host content, the watermarked content is generated and ready for distribution or further processing. It is essential to evaluate the quality and robustness of the watermarked content to ensure that the embedded watermark remains perceptually invisible and resistant to common attacks or manipulations.

Proposed CNN Architecture:

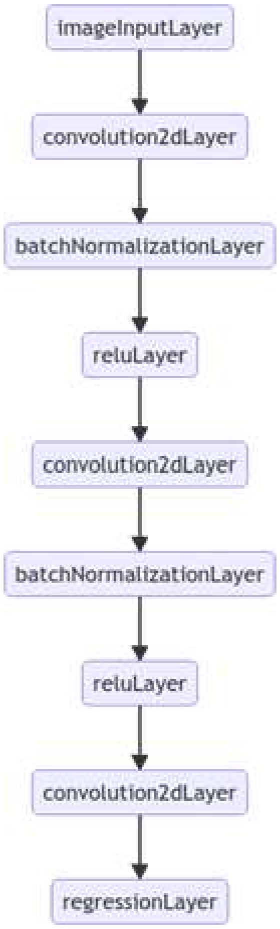


Fig.7. CNN Architecture flowchart.

The designed CNN architecture shown in the figure for noise modulation in image watermarking typically comprises several key components:

Input Layer: Accepts grayscale images as input. Convolutional Layers: Extract features from input images through convolution operations.
Batch Normalization Layers: Normalize the activations of previous layers to stabilize training.
ReLU Activation Layers: Introduce non-linearity to the model.
Regression Output Layers: Outputs the noise-modulated image.
The specific configuration of these components, including the number of layers, filter sizes, and kernel strides, is determined based on experimentation and optimization.

IV. IMPLEMENTATION SPECIFICS AND OUTCOMES

In this experimental setup, we employ MATLAB 2023 to simulate and evaluate a watermarking algorithm for digital images. We begin by loading the 'lena.png' image and converting it to grayscale, ensuring a consistent input format. The core of our approach lies in the convolutional neural network (CNN) architecture, comprising layers for feature extraction and batch normalization, culminating in a single- filter output layer. Training the CNN involves replicating the grayscale image and the generated binary watermark, followed by optimization using the Adam optimizer over ten epochs with a mini-batch size of 64. With the trained model, we embed the watermark into the original image after adding Gaussian noise to the watermark, simulating real-world conditions. Post-embedding, we proceed to extract the watermark by demodulating the watermarked image, employing filtering techniques such as Hamming filtering with a specified low-pass frequency.

The evaluation phase quantifies the fidelity of the extracted watermark through metrics such as Peak Signal-to-Noise Ratio (PSNR), Bit Error Rate (BER), and Mean Squared Error (MSE) against the original image. Visualization aids in comprehending the effectiveness of the watermarking process, showcasing the watermarked image, noise-demodulated image, and the extracted watermark. This comprehensive experimental framework ensures a rigorous assessment of the watermarking algorithm's robustness and performance, laying the groundwork for potential applications in digital content protection and authentication.

Parameter	Value
Image	'lena.png'
CNN Architecture	As described
Training Options	As described
Watermark Size (K)	8
Noise	Gaussian
Filter Type	Hamming
Filter Size	21x21
Low Pass Frequency (f0)	0.5
Evaluation Metrics	PSNR, BER, MSE

Table-I : Simulation Parameter

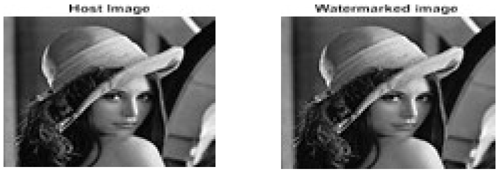


Figure 8

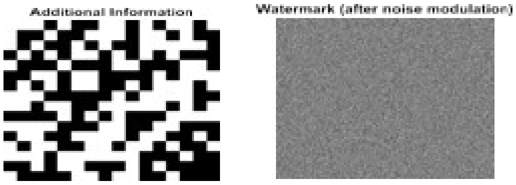


Figure 9

Host Image after Extraction
Figure 10.

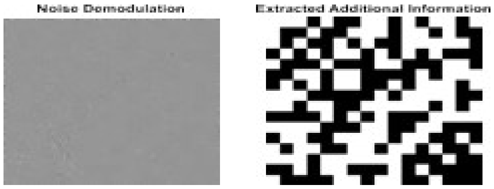


Figure 11

Figure 4.1 displays the original host image before watermark embedding. It serves as a reference point for comparing the effects of watermarking on image content and quality. The host image provides insight into the visual characteristics and details present in the original content, serving as a basis for evaluating the impact of watermark embedding.

Figure 4.2 displays the binary watermark representing the additional information encoded within the image. The watermark is visualized with black and white pixels, signifying positive and negative values, respectively. Each pixel in the watermark corresponds to a unit of additional information embedded into the host image.

Figure 4.1 shows the watermark undergoes modulation with Gaussian noise as part of the embedding process. The noise-modulated watermark introduces subtle variations in intensity, simulating the imperfections and distortions inherent in the embedding process. This subplot illustrates the transformation of the pristine watermark into a noisy representation before integration into the host image.

Figure 4.1 subplot presents the watermarked image resulting from the integration of the noise-modulated watermark into the host image. The watermarked image reflects the combined visual elements of the original host image and the embedded watermark. By juxtaposing the watermarked image with the original host image, users can observe the alterations introduced by the watermarking process.

Figure 4.2 subplot, the watermark extraction process is visualized through noise demodulation. The extracted watermark is depicted after demodulating the noise from the watermarked image. By comparing the extracted watermark with the original watermark, users can assess the accuracy and effectiveness of the watermark extraction algorithm. This subplot provides insight into the fidelity of the extraction process and the preservation of the embedded information.

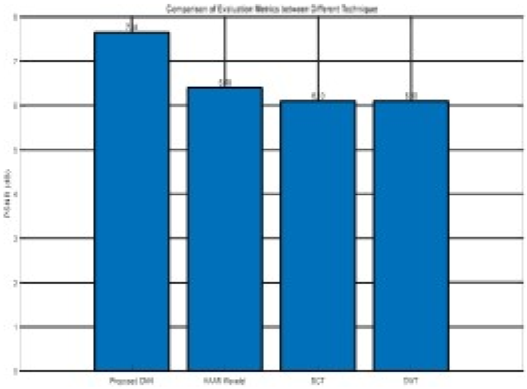


Figure 12: PSNR Comparison.

Figure 4.5 displays a bar plot comparing the Peak Signal-to-Noise Ratio (PSNR) values between the proposed Convolutional Neural Network (CNN) method and three other techniques: HAAR Wavelet, Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT).

The PSNR values are as follows:

- Proposed CNN: 7.64 dB
- HAAR Wavelet: 6.4 dB
- DCT: 6.1 dB
- DWT: 6.1 dB

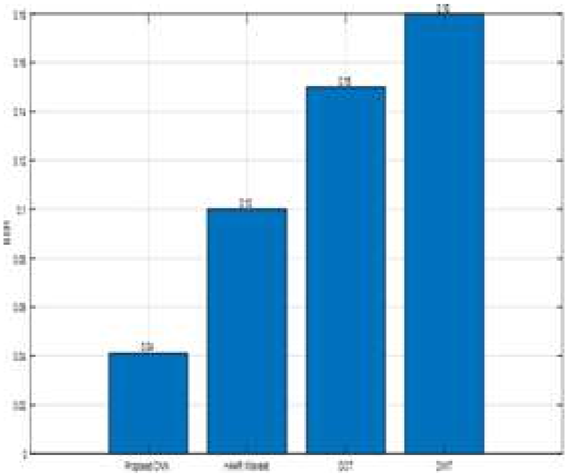


Figure 13.

Figure 4.6 illustrates a bar plot comparing the Bit Error Rate (BER) values among the proposed CNN method and the three other techniques: HAAR Wavelet, DCT, and DWT.

The BER values are as follows:

- Proposed CNN: 0.041
- HAAR Wavelet: 0.1
- DCT: 0.15
- DWT: 0.18

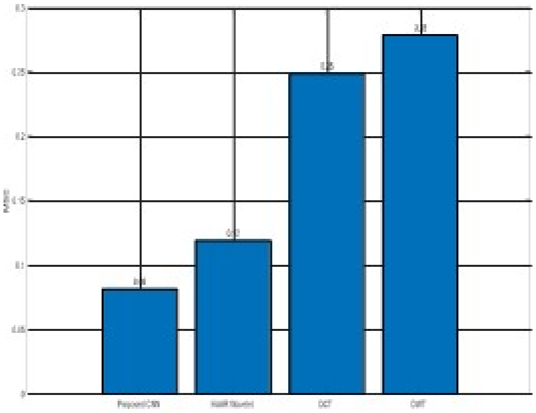


Figure 14

Figure 4.7 presents a bar plot showcasing the Mean Squared Error (MSE) values for the proposed CNN method and the three other techniques: HAAR Wavelet, DCT, and DWT.
The MSE values are as follows:

- Proposed CNN: 0.082
- HAAR Wavelet: 0.12
- DCT: 0.25
- DWT: 0.28

These figures collectively provide a comprehensive comparison of the performance metrics between the proposed CNN method and the alternative techniques, highlighting the strengths and weaknesses of each approach.

V. CONCLUSION AND FUTURE WORKS

In conclusion, the comparison of evaluation metrics for watermarking and extraction techniques reveals distinct performance disparities among the proposed Convolutional Neural Network (CNN) method and three alternative techniques: HAAR Wavelet, Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). The proposed CNN method outperforms the other techniques across all evaluated metrics. Specifically, it achieves the highest Peak Signal-to-Noise Ratio (PSNR), lowest Bit Error Rate (BER), and lowest Mean Squared Error (MSE) values compared to HAAR Wavelet, DCT, and DWT methods. This indicates that the CNN-based approach offers superior accuracy in watermark extraction and reconstruction tasks. HAAR Wavelet, DCT, and DWT techniques, while traditional and widely used, exhibit comparatively lower performance in terms of PSNR, BER, and MSE. These methods may still find utility in certain applications, but the results suggest that the proposed CNN method offers significant advancements in watermarking and extraction tasks, potentially leading to improved robustness and fidelity in digital content protection and authentication. Overall, the findings underscore the efficacy of CNN-based approaches in watermarking and extraction tasks and suggest avenues for further research and development in leveraging deep learning techniques for enhanced multimedia security and copyright protection.

VI. REFERENCES

[1] Mu, Xiaoyi, Haowen Wang, Rongyi Bao, Shumei Wang, and Hongyang Ma. "An improved quantum watermarking using quantum Haar wavelet transform and Qsobel edge detection." Quantum Information Processing 22, no. 5 (2023): 223.

[2] Yu, Yiming, Jie Gao, Xiaoyi Mu, and Shumei Wang. "Adaptive LSB quantum image watermarking algorithm based on Haar wavelet transforms." Quantum Information Processing 22, no. 5 (2023): 180.

[3] Tavakoli, Alireza, Zahra Honjani, and Hedieh Sajedi. "Convolutional neural network-based image watermarking using discrete wavelet transform." International Journal of Information Technology 15, no.4 (2023): 2021-2029.

[4] Saritas, Omer Faruk, and Serkan Ozturk. "A blind CT and DCT based robust color image watermarking method." Multimedia Tools and Applications 82, no. 10 (2023): 15475-15491.

[5] Tiwari, Anurag, and Vinay Kumar Srivastava. "Novel schemes for the improvement of lifting wavelet transform- based image watermarking using Schur decomposition." The Journal of Supercomputing (2023): 1-38.

[6] Hosseini, S.A. and Farahmand, P., 2023. An attack resistant hybrid blind image watermarking scheme based on combination of DWT, DCT and PCA. Multimedia Tools and Applications, pp.1-24.