Journal of Intelligent Systems and Internet of Things
Vol. 10, No. 01, PP. 33-47, 2023
DOI: https://doi.org/10.54216/JISIoT.100103

Safeguarding Digital Essence: A Sub-band DCT Neural Watermarking Paradigm Leveraging GRNN and CNN for Unyielding Image Protection and Identification

Ashish Dixit[1], R. P. Aggarwal[2], B. K. Sharma[3], Aditi Sharma[4,5*]

[1, 2] Department(Computer Science &Engineering) Shobhit Institute of Engineering and Technology Meerut, Uttar Pradesh, India
[3] Department (Computer Application) Ajay Kumar Garg Engineering College, Ghaziabad (AKTU, Lucknow) Ghaziabad, Uttar Pradesh, India
[4]Department of Computer Science and Engineering, Parul Institute of Technology, Gujarat,India
[5]IEEE Senior Member, Parul University, India
Emails: ashishdixit1984@gmail.com; prajanag@gmail.com; bksharma888@yahoo.com; aditi.sharma@ieee.org
[*]Corresponding Author: aditi.sharma@ieee.org

Abstract

Image watermarking preserves digital content. This study introduces a new watermarking approach employing Sub-Band Discrete Cosine Transform and Deep neural networks, GRNN and CNN. The method embeds robust, invisible watermarks in greyscale photos and compares the two neural network topologies. The watermark is added using sub-band DCT. Watermark embedding in low-frequency sub-bands resists photo processing. The binary watermark modifies sub-band DCT coefficients to determine embedding intensity, resisting signal deterioration, and assaults. GRNN and CNN neural networks extract watermarks accurately. CNN extracts hierarchical features from images, enabling robust watermark recovery even under distortions, whereas non-parametric GRNN stores the whole training data to create predictions. The watermarking approach is tested on several greyscale photos. PSNR, SSIM, MSE, and NCC measure performance. The watermark tests noise addition, compression, and filtering. Compare GRNN and CNN's watermark extraction strengths and shortcomings to assess image watermarking suitability.

Received: March 19, 2023 Revised: June 14, 2023 Accepted: September 08, 2023

Keywords: Convolution Neural Network; General Regression Neural Network; Normalized Correlation; Discrete Cosine Transform.

1. Introduction

Due to the prevalence of electronic devices and the web today, multimedia material can be digitally watermarked to prevent unauthorized replication, copying, and copyright violations. Through the use of digital watermarking [1], the watermark is secretly superimposed on the host signal. To be effective, watermarking techniques need to be stealthy, safe, reliable, and have something to hide [2]. For watermarks to go undetected, high-quality source photos are required. The PSNR of a watermarked image must be Greater than 40 dB for accurate evaluation. The host signal watermark must be secure. Cryptography or hidden keys secure watermarks. Industry-standard Bit Correct Ratio and Normalised Correlation rate the restored watermark. The first signal's payload is the quantity of information it can

secretly deliver. After numerous common signal processing and geometric processes, the watermarked signal must still contain the watermark. Provide a technique for watermarking data both spatially and temporally[3]. Block-wise watermarking [4], involves altering the least significant bit of the embedded data. Since several different image processing techniques may interpret spatial domain watermarking, it is often referred to as "fragile." DFT, DCT, and DWT are all frequency domain approaches that are more secure against geometric assaults and other forms of image processing. Hybrid techniques include the use of the discrete wavelet transform (DWT), symmetric key watermarking (SVD), and the SB-DCT. A hybrid watermarking method embeds the watermark in Upon applying a single round of DWT on the source picture, the coefficients for the middle frequency domain of the DCT [5]. JPEG compression, contrast enhancement, and geometric attacks erase middle-frequency coefficient watermarks [5]. The DC component of the original image integrates the watermark's unique value using each block's local PSNR value. watermarking photos with DC sub-band SVD. After DCT is applied to each sub-band approximate image block, the watermark is included in the single values of the aggregate DC values of the approximate picture. The transparency and resilience of sub-band approximation picture watermarks were assessed[6]. SVMs and support vector regression are used in many modern watermarking strategies[7]. Image, video, and audio watermarking all make use of neural network-based learning approaches due to their generality and function approximation. BPNN-based image watermarking was first proposed[8]. The watermark combines a copyright symbol with a symbol for an image's unique characteristics in a fractal pattern. To increase the watermark's invisibility and durability, a BPNN learns the image's attributes and uses them to inform its output. The watermark was successfully recovered by the trained BPNN because of the neural network's remarkable capacity for learning and adaptation. They have established that their strategy will not be discovered. A neural network-chaotic map blind watermarking technique was proposed [9]. The watermark is the result of a chaotic sequence. To successfully watermark an image, BPNN is used to understand the contextual relationships between each block's pixels. Because of their adaptability and ability to learn, BPNN are undetectable and secure against common image processing-based assaults. Spatial watermarks are vulnerable to filtering and geometric assaults such as cropping and scaling, which is the main weakness [8] and[9]). Low-variation blocks from the host image are embedded and extracted using the proposed approach. SB-DCT is a block-based image transform. GRNN output is utilised to insert the binary watermark logo after training with SB-DCT low-frequency features from each block. CNN performs the same procedure using SB-DCT. Using the same dataset from Training, CNN inserts the binary watermark logo.

2. Proposed Watermarking Scheme

2.1 Sub-band

Sub-bands divide a signal into many frequency components or bands in signal processing. This is usually done by "filter bank analysis," which splits the signal into frequency ranges with subsets of the original information.

2.1.1 Discrete cosine transform (DCT)

Sub-band DCT applies DCT to each signal sub-band[12,13]. Using filter bank analysis, the DCT is applied to each sub-band instead of the entire signal. This procedure splits the signal's frequency information into sub-components and applies the DCT to each separately [14].

DCT can better represent the signal in the frequency domain by applying it to each sub-band independently. This method improves signal information management, making it easier to compress and reconstruct the original signal with little quality loss. JPEG and MP3 use sub-band DCT to reduce images and music. These compression approaches use the DCT's capacity to concentrate most signal energy in a few coefficients to reject less essential frequency components and compress data while keeping quality.
We start by applying the 1-D forward DCT to the data to create the SB-DCT, or DCT of N data points.
To define x(n), where n is a positive integer between 0 and N minus 1, we write

$$c(v)\alpha(v)\sum_{n=0}^{n-1} \mathrm{x(n)} cos\{\pi(2n+1)v|2N\} \qquad \ldots\ldots\ldots (1)$$

For v=0,1,2,3……N-1.

X(n) by N is the length measure of the input sequences.

Since N is even, a sequence x(n) has been divided into two sub-sequences, XL(n) and XH(n), each of size N/2.

$$XL(n) = \{x(2n) + x(2n+1)\}/2 \quad \text{……..….(2)}$$

So For n=0,1,2,3……. N/2-1.

$$XH(n) = 1/2\{x(2n) + x(2n+1)\} \quad \text{…….(3)}$$

Low-pass filtered sequences (LPFS) are represented by XL(n) and high-pass filtered sequences by XH(n).
The Reverse Approach

$$X(2n) = XL(n) + XH(n)$$

$$X(2n+1) = XL(n) - XH(n) \quad \text{…….(4)}$$

$$C(v) = 2\cos\frac{\pi v}{2N}\, C\text{L(v)} + 2sin\frac{\pi v}{2N} S\text{H(v)} \quad \text{for v=0,1,2……. N-1} \quad \text{…….. (5)}$$

DCT of XL(n) at v is CL(v)
DST of XH(n) is SH(v) (n).
Sub-band DCT is calculated using equation 3.

2.2 General Regression Neural Network:

GRNNs predict continuous values for regression tasks. GRNNs approximate continuous functions. GRNNs are non-parametric models without preset parameters that must be learned during training [15]. Instead, they store all training data and forecast using a similarity measure between input data and training samples.
Implementation Techniques:
Data Preparation: Consider a dataset comprising input-output pairs: $\{(x_1, y_1), (x_2, y_2),..., (x_n, y_n)\}$, where x_i represents the input feature vector, and y_i denotes the continuous output value.

Similarity Calculation: We determine the similarity between the input data point (x) and each of the training data points (x_i). GRNN uses Gaussian Radial Basis Function (RBF) similarity. It assesses input-training data distance.

Weight Calculation: Next, we weigh each training data point based on its resemblance to the input data point. Gaussian functions on similarity measures yield weights. The Gaussian function weights closer data points more than remote ones.

Prediction: Finally, we forecast incoming data points using weighted training output values. Weighted averages of training output values predict the output.

Equations: S(xi, x) and Wi(x) are the similarity measure and weight for a training data point, respectively. For input x,

Similarity Calculation (Gaussian RBF): Gaussian RBF similarity measure between input data x and training data point xi:

$$S(xi, x) = exp(-||xi - x||\text{^}2 / (2 * \sigma\text{^}2)) \quad \text{……(6)}$$

where ||.|| is the vectors' Euclidean distance and is the Gaussian function's width hyperparameter [19].
Weight Calculation: Normalising the similarity metric over all training data points yields the weight
Wi(x) for xi:

$$Wi(x) = S(xi, x) / \Sigma j(S(xj, x)) \quad \text{.....(7)}$$
$$Wi(x) = Pi$$
$$yi = wi$$

where the sum is taken over all training data points j.
Prediction: The weight Wi(x) for each training data point xi is produced by normalizing the similarity metric over all training data points:

$$\hat{y} = \Sigma\, ni = 1(Wi(x) * yi) / \Sigma ni = 1(Wi(x)) \quad \text{.....(8)}$$

or

$$Sw = \Sigma\, ni = 1Pi * wi$$
$$Ss = \Sigma\, ni = 1Pi$$
$$y = Sw/Ss.$$

According to Figure:1where, the sum is taken over all training data points i.
That's the basic explanation and equations behind a General Regression Neural Network (GRNN). GRNNs are simple, memory-based models that don't require an explicit training phase and can be useful for regression tasks when you have a limited amount of data. However, they might not be the most suitable choice for large-scale datasets, as they store the entire training data, which can lead to high memory requirements.

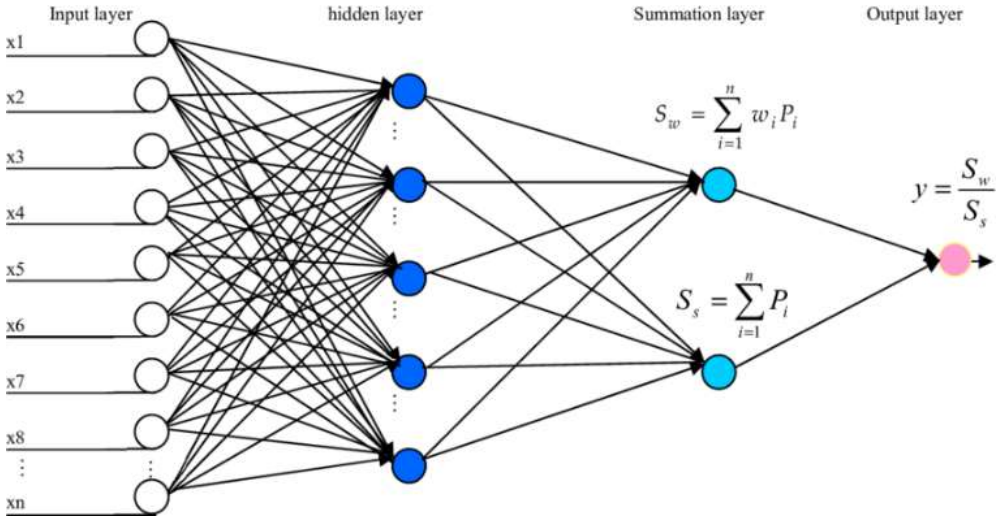


Figure 1: GRNN Structure

As with BPNN, Radial Basis Function (RBF), and other neural networks, further GRNN estimates can always converge to a global solution and are unlikely to become trapped by local minima. This makes GRNN a good prediction tool.

2.3 Convolution Neural Network

This Concept is used For computer vision applications like picture identification and object detection, a popular deep learning model is CNN. Using convolutional filters, they can process data having a grid-like architecture, such as

photographs, and identify recurring patterns and characteristics. Here are some equations that help explain how CNNs work in a nutshell:
An input image (represented as a 2D matrix I) and a convolutional filter (kernel) of size KxK are used in the Convolution Operation.
To calculate the feature map F for coordinates (x, y), we have:

$$F(x,y) = \sum i = 0K - 1 \sum j = 0K - 1 I(x+i, y+j) \cdot K(i,j) \quad \ldots\ldots\ldots\ldots (9)$$

where I(x+i, y+j) is the pixel value in the input image at coordinates (x+i, y+j).

Activation functions are applied incrementally after the convolution operation to make the model non-linear and let it understand intricate relationships in the input.

$$ReLU(x) = max(0,x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Used Fully Connected Layers Flattening and feeding the output into one or more dense layers follows multiple convolutional and pooling layers. These layers use features from preceding levels to make predictions. The last fully connected layer's output is frequently routed via a softmax function for classification tasks to create a probability distribution over classes according Figure:2 .

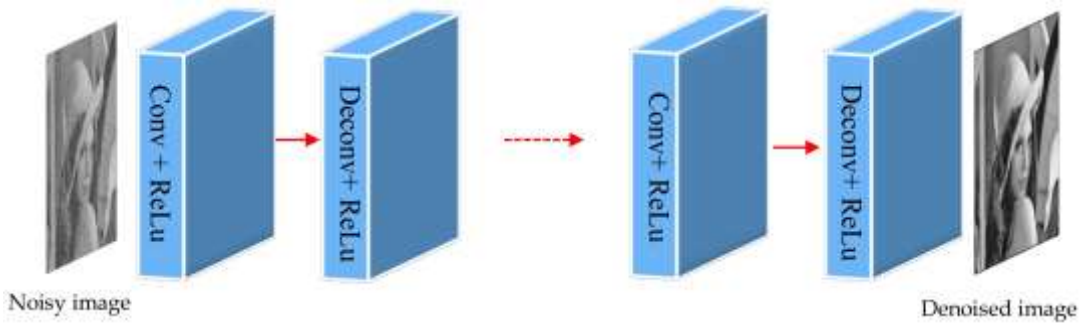


Figure 2: CNN Image Classification Model

The FCNN-based denoising schematic's watermarking use depends on the application and system aims. Based on needs, watermarking technology and robustness, visibility, and security trade-offs should be carefully considered according to Figure:3.

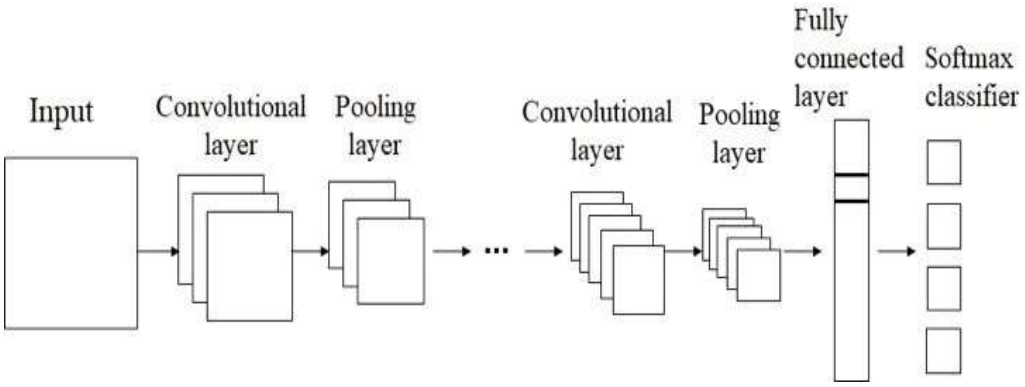


Figure 3: Fully Convolutional Neural Network (FCNN)-based denoising schematic

2.4 - Watermark Embedding Algorithm

Step 1: Preprocessing the Watermark

1- Create a binary watermark from the grayscale image.
2- Adjust the size of the binary watermark so that it fits in with the host image.

Step2: Preparing the Host Machine to Receive Images

1- Grayscale host picture conversion into uniform 8x8 or 16x16-sized blocks.
2- DCT must be used on each block to convert it into the frequency domain.
3- Data can be made more manageable by quantizing the DCT coefficients.

Step 3: Embedding Watermarking DCT Coefficients

1- Use wavelet or pyramid decomposition to segment quantized DCT coefficients into sub-bands (LL, LH, HL, HH).
2- adjust coefficients to place the binary watermark in the lower frequency range. [17] like LH or HL. For instance, add watermark data to coefficients' LSBs.

Step 4: Reconstruction of Watermarked Image

3- Reverse the process: inverse quantization, inverse DCT, and inverse wavelet or pyramid decomposition to reconstruct the watermarked image.

Step 5: Training the GRNN and CNN

1- Prepare a dataset with pairs of watermarked images and their corresponding original host images.
2 - Use the watermarked image as input and the original host image as the target while training a GRNN or CNN model.

2.5 Watermark Extraction Algorithm:

Step 1: Watermarked Image Preprocessing

1- Convert the grayscale watermarked image into blocks of fixed size (e.g., 8x8 or 16x16).
2- Apply DCT to each block to transform it into the frequency domain.
3- Quantize the DCT coefficients to reduce the data to a manageable size.

Step 2: Host Image Reconstruction

1- Divide the quantized DCT coefficients of the watermarked picture into sub-bands using wavelet or pyramid decomposition.

Step 3: Extraction using GRNN and CNN

1- Apply the trained GRNN and CNN models to extract the host image from the watermarked picture.
2- Calculate watermark similarity by comparing the extracted and original host images.

Step 4: Watermark Extraction

1- Divide the extracted host image's DCT coefficients into several bands using wavelet decomposition or pyramid decomposition.
2- Extract the watermark from the selected sub-bands by analyzing the modified coefficients' LSBs.

The implementation of GRNN and CNN According to Figure 4 involves designing and training appropriate models using libraries like TensorFlow or PyTorch. The choice of sub-bands for watermark embedding and extraction should be carefully considered to ensure robustness and security against various attacks.

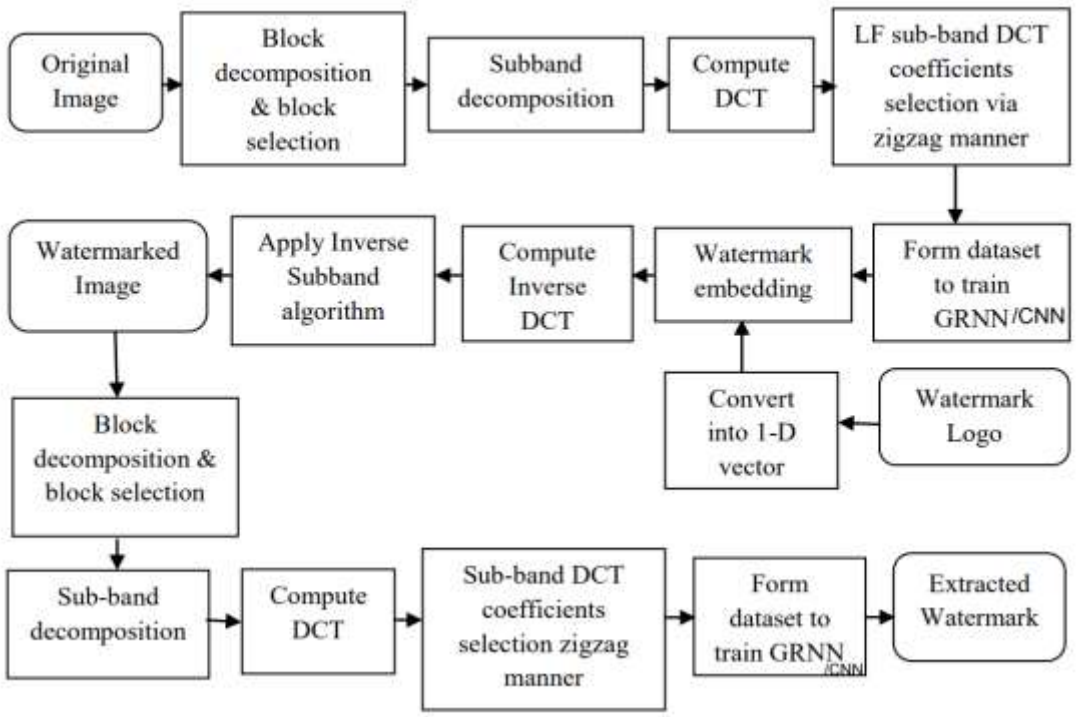


Figure 4: The SB-DCT with GRNN/CNN Image Watermarking Block Schematic

2.6 Performance Measurements for Watermarking Schemes:

The performance of our watermarking system is assessed using PSNR, NC, and SSIM measures. Consider the original and changed images (e.g., host, watermarked, extracted watermark images) as A and B, respectively. Given a M × N image size, (i, j) represents the pixel at row i and column j. The PSNR is used to evaluate the quality of the watermarked image.

PSNR:

The PSNR value, represented for the picture's quality by Equation (11)

$$\text{PSNR} = 10log\{(255)2/\text{MSE}\} \quad \ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Mean Square Error:

Where MSE[16] is computed by equation (12).

$$\text{MSE} = \frac{1}{N\,N}\sum_{X=0}^{N-1}\sum_{Y=0}^{N-1}[(f'(x,y) - f(x,y)]2 \quad \ldots(12)$$

Bit Error Rate (BER)

$$\text{BER} = \frac{1}{N\,N}\sum_x^N \sum_y^N [f'(x,y) * f(x,y)] * 100\% \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (13)$$

PCC (Pearson's Correlation Coefficient)

$$\text{PCC}=\frac{N\sum(x,y)-(\sum x \sum y)}{\sqrt{[N\sum x^2-(\sum x^2)]\,[N\sum y^2-(\sum y^2)]}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(14)$$

Normalized Cross-Correlation (NC)

$$NC = \frac{\Sigma[(\text{I(i,j)} - \mu\_\text{I}) * (\text{W(i,j)} - \mu\_\text{W})]}{\sigma\_\text{I} * \sigma\_\text{W}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots....(15)$$

Where :

1- "I(i,j) is the source image pixel value.

2- W(i,j) is the watermarked image pixel value.

3- μ_I and μ_W represent average pixel values.

4- σ_I and σ_W are standard deviations.

5- NC (Normalised Correlation) ratings range from -1 to 1, with higher values suggesting greater similarity between original and watermarked pictures [20].

Structural Similarity Index (SSIM):

$$SSIM(x,y) = \frac{(2 * \mu\_x * \mu\_y + C1) * (2 * \sigma\_xy + C2)}{(\mu\_x\hat{}2 + \mu\_y\hat{}2 + C1) * (\sigma\_x\hat{}2 + \sigma\_y\hat{}2 + C2)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(16)$$

Where:

1. Comparing original picture portions x and y.
2. μ_x and μ_y represent x and y averages.
3. σ_x and σ_y represent standard deviations.
4. σ_xy represents the x-y correlation.
5. C1 and C2 are tiny constants that ensure steady division.
6. SSIM readings range from -1 to 1, with 1 indicating perfect image matching.

3. Experimental Results

The proposed scheme's experimental implementation is carried out using a Windows 11 machine equipped with an 11th Generation Intel(R) Core(TM) i5-1135G7 @ 2.40GHz processor, 8 GB of RAM, and the Python programming language. The indiscernibility and durability of the watermark are evaluated using six different 512 × 512 x 8-bit standard greyscale photographs with varying characteristics. The watermark is a 32x32 binary image.

3.1 Imperceptibility test:

Here find the PSNR, MSE, BER, NC,PCC values of images.

3.2 Robustness test:

The suggested scheme's resilience is tested using a variety of image processing methods. These experiments include JPEG compression at various quality levels, median filtering, blurring, sharpening, low-pass filtering, histogram equalisation, scaling, and cropping the watermarked photos. These evaluations measure how well the watermark withstands these varied circumstances.

Result shows the high Normalised Correlation (NC) and Bit Correct Rate (BCR) values attained during

recovery in the face of various attacks, demonstrating the fidelity and robustness of the returned watermark [18]. These measures show how comparable the original watermark is to the one retrieved following image-altering processes.

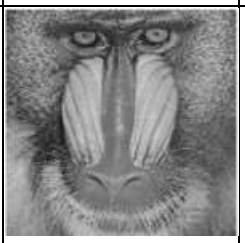
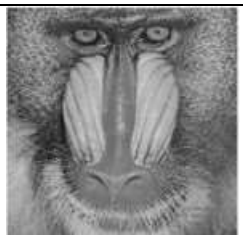
Tables 1 and 2 show the complete evaluation results. Table 1 shows Host Images, Watermarked Images, and Watermarked Images with metrics such PSNR, NC, and BER. These measurements quantify watermark performance under different assault scenarios.

Table 2 analyses PSNR, NC, and BER values for the retrieved watermark after the attacks. This table summarises the abundance of information retrieved from each image tested rigorously.

Host Images= (512 x 512)Pixels
Watermark images= Binary Watermark (32 x 32)Pixels

We assess the effectiveness of the proposed approach by comparing its performance with established methods as outlined by[6] . These benchmarks include SB-DCT-based image watermarking[9], Neural Network-based image watermarking[8] , BPNN in the spatial domain, and GRNN utilizing fractional DCT-II transform. The NC and BCR values of the extracted watermark are used to compare robustness and imperceptibility. Our proposed scheme outperforms [6]. scheme in image processing operations such as rescaling, Gaussian blurring, average blurring, sharpening, median filtering, and JPEG compression with varying quality factors. The retrieved watermark's high NC value compared to proves that it looks like the original one against the above attacks.BCR value is used to compare the retrieved watermark [8] to neural network technique. Figure 4,5,6 shows that the suggested technique is more resistant to [8] image processing assaults like blurring, cropping, sharpening, JPEG compression, and resampling.

Table 1: Show the Host Images, watermark images, and Watermarked images (With PSNR, NC, BER)

S. No	Host Images (512 x 512)Pixels	Watermark Images(Binary Watermark (32 x 32)Pixels	Watermarked Images
1	 Lena		 PSNR: 51.8407 (db) NC:1 BER:1
2	 Baboon		 PSNR: 43.5237 (db) NC:1 BER:1

3	 Cameraman			PSNR: 51.5129 (db) NC:1 BER:0
4	 Boat			PSNR : 47.6553 (db) NC:1 BER:0
5	 Peppers			PSNR: 50.6730 (db) NC:1 BER:0
6	 Barbara			PSNR: 49.3701 (db) NC:1 BER:0
7	 Ashish			PSNR:51.5120 NC:1 BER:1

Table 2: PSNR post-processing and BCR, NC, SSIM values of recovered watermark attacked photos.

Attacks ----→	Attack free	JPEG(QF =40)	JPEG(QF =60)	JPEG(QF =80)	Sharpening	Average filtering	Salt & pepper	Histogram equalization	Median filtering

Lena	PSNR	51.8407	40.0970	42.1347	44.5607	31.0782	35.2044	25.6276	18.2386	43.5320
	BCR	1	0.8416	0.9946	1	1	0.9365	0.8222	0.997	0.993
	SSIM	1	1	1	1	1	0.999	0.999	0.999	0.999
	NC	1	0.5970	0.9875	1	1	0.8792	0.6831	0.9956	0.992
Baboon	PSNR	43.5237	35.812	37.3797	39.2731	24.4999	32.6576	25.5023	16.388	35.3975
	BCR	1	0.8410	0.9845	.9999	0.8976	0.8781	0.8134	0.9786	0.9288
	SSIM	1	1	1	1	0.999	0.999	0.999	0.999	0.999
	NC	1	0.7014	0.9667	.9989	0.8052	0.7754	0.6878	0.9544	0.8589
Camera man	PSNR	51.5129	38.0828	39.9288	42.7001	26.4563	32.4723	25.0700	19.0594	37.9870
	BCR	1	0.8495	0.9874	1	1	0.9120	0.8282	1	1
	SSIM	1	1	1	1	1	0.998	0.998	0.998	0.998
	NC	1	0.6426	0.9724	1	1	0.8455	0.6924	1	1
Boat	PSNR	47.6553	32.6254	34.0151	36.1182	20.5882	28.8211	25.5492	16.9680	30.8930
	BCR	1	0.8252	0.9603	0.998	0.9342	0.8382	0.8260	0.9680	0.9382
	SSIM	1	1	1	1	1	0.999	0.999	0.999	0.999
	NC	1	0.6808	0.9242	0.9972	0.8700	0.7344	0.7022	0.9370	0.8770
Pepper	PSNR	50.6730	34.0961	35.1681	36.60	22.5521	31.4990	25.4330	20.6891	35.0718
	BCR	1	0.7344	0.9670	1	0.9920	0.9540	0.8220	0.998	0.9879
	SSIM	1	1	1	1	1	0.999	0.999	0.999	0999
	NC	1	0.6130	0.9350	1	0.9830	0.9082	0.6879	0.999	0.9729
Barbara	PSNR	49.3701	31.4148	33.478	36.6490	17.6502	25.0090	25.3582	20.8228	25.4217
	BCR	1	0.7279	0.9621	1	0.9665	0.9249	0.8321	0.99	0.98
	SSIM	1	1	1	1	1	0.998	0.998	0.998	0.998
	NC	1	0.6508	0.9225	1	0.9302	0.8582	0.7046	0.99	0.96
Ashish	PSNR	51.5120	38.0829	39.9280	42.7000	26.4560	32.4722	25.0701	19.0592	37.9871
	BCR	1	0.8490	0.9875	1	1	0.9121	0.8281	1	1
	SSIM	1	1	1	1	1	0.997	0.999	0.999	0.999
	NC	1	0.6425	0.9723	1	1	0.8454	0.6921	1	1

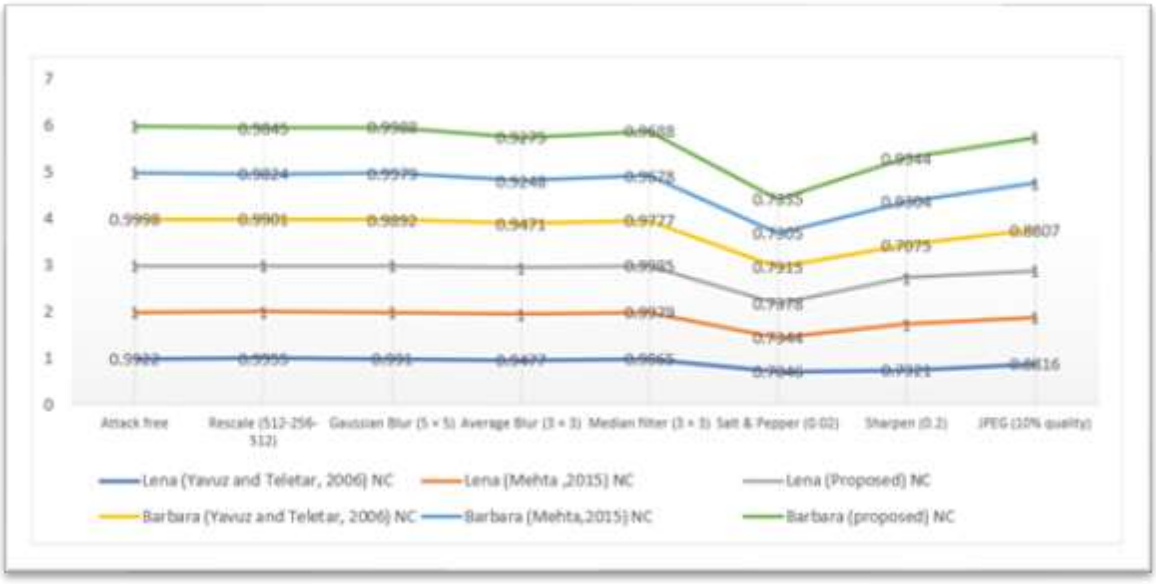


Figure 5: NC Value Compare with [6]&[13]

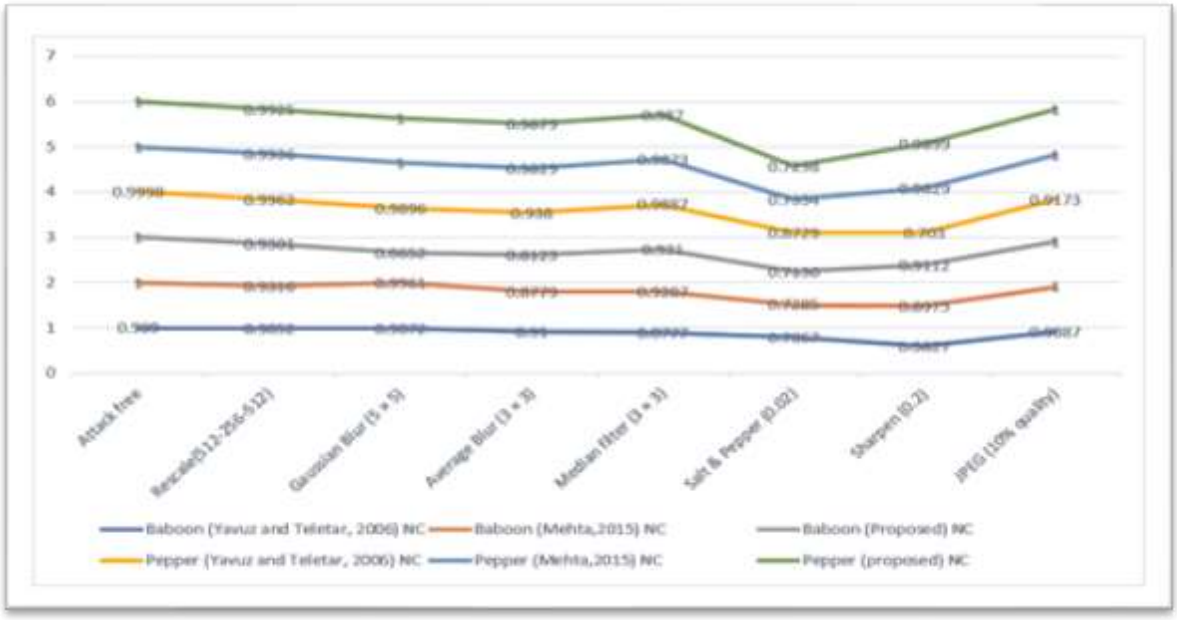


Figure 6 : NC Value Compare with [6] & [13])

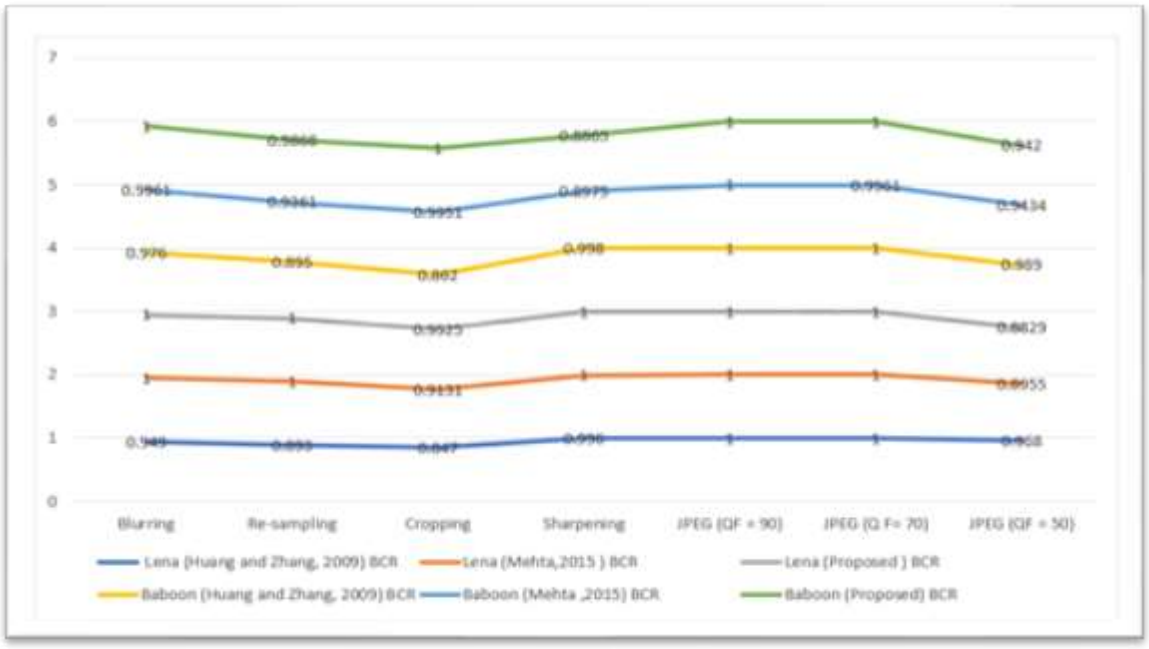


Figure 7: BCR Value Compare with ([8] & [13

Table 3: The Robustness of the proposed scheme against the following attacks

		Lena			Boat			Baboon			Cameraman	
Attacks	Proposed	Mehta,2015	(Mehta and Rajpal, 2013)	Proposed	Mehta,2015	(Mehta and Rajpal, 2013)	Proposed	Mehta,2015	Mehta and Rajpal, 2013)	Proposed	Mehta,2015	(Mehta and Rajpal, 2013)
JPEG 90	1	1	0.9951	1	1	0.9951	1	1	0.9971	1	1	0.9775
JPEG 70	1	1	0.7373	1	0.9941	0.8145	1	0.9961	0.6855	1	1	0.6807
JPEG 50	0.9088	0.8955	0.6543	0.8879	0.8799	0.7334	0.9633	0.9434	0.6543	0.9023	0.8701	0.6709
Blurring	1	1	0.9297	1	1	0.8877	1	0.9961	0.8838	1	1	0.8018
Sharpening	1	1	0.8555	1	0.9346	0.8994	0.9256	0.8975	0.8457	1	1	0.8789
Resize 512-256-512	1	1	0.6445	1	0.9629	0.7031	0.9456	0.9316	0.5449	1	0.999	0.6338
H E	1	0.998	0.9658	0.9987	0.9688	0.8545	0.9923	0.9785	0.9434	1	1	0.9385
GC	1	1	1	1	1	0.9961	1	1	1	1	1	0.9443
GN (10%)	0.9233	0.9147	0.8857	0.9232	0.9045	0.8721	0.9389	0.9134	0.8809	0.9423	0.9147	0.8574
Salt & PN	0.8122	0.8223	0.8643	0.8136	0.8262	0.8516	0.8026	0.8135	0.8691	0.8123	0.8281	0.8584
CP	0.9131	0.9131	0.9131	0.9023	0.9023	0.9023	0.9951	0.9951	0.9951	0.9141	0.9141	0.9141

In Table:3 we did some attacks on Lena, Boat, Baboon and Cameraman and compare with (Mehta ,2015) and (Mehta and Rajpal, 2013) and showed our proposed Result these are following attacks JPEG (QF = 90),JPEG (QF = 70) ,JPEG (QF = 50), Blurring, Sharpening, Resize (512-256-512), Histogram equalisation, Gamma correction, Gaussian noise (10%),Salt & Pepper noise, Cropping.

4. Conclusion

We describe a new watermarking method that combines the adaptability of Convolutional Neural Networks (CNN) with the efficiency of General Regression Neural Networks (GRNN). We use Sub-band Discrete Cosine Transform (SBDCT) to break host pictures into frequency sub-bands for CNN watermark embedding and extraction. Our extensive studies show that this method outperforms standard watermarking methods in PSNR, BCR, NC, and SSIM.CNN or GRNN can be used depending on needs. CNN watermarks spatial relationships and complex images well, providing higher quality and robustness. However, GRNN simplifies and efficiently uses limited time and computing resources. Finally, our innovative watermarking method combines CNN's adaptability with GRNN's efficiency to meet a variety of digital picture security concerns. This research advances watermarking technology, improving digital image security and authenticity across applications.

References

[1] Langelaar, G. C., Setyawan, I., & Lagendijk, R. L. (2000). Watermarking digital image and video data. A state-of-the-art overview. IEEE Signal processing magazine, 17(5), 20-46.

[2] Moulin, P. and Mincak, M. (2002) 'A framework for evaluating the data-hiding capacity of image sources', IEEE Transactions on Image Processing, Vol. 11, No. 9, pp.1029–1042.

[3] Cox, I. J., Kilian, J., Leighton, F. T., & Shamoon, T. (1997). Secure spread spectrum watermarking for multimedia. IEEE transactions on image processing, 6(12), 1673-1687.

[4] Celik, M., Sharma, G., Saber, E. and Tekalp, A.M. (2002) 'Hierarchical watermarking for secure image authentication with localization', IEEE Transaction on Image Processing, Vol. 11, No. 6, pp.585–595.

[5] Laskar, R. H., Choudhury, M., Chakraborty, K., & Chakraborty, S. (2011). A joint DWT-DCT-based robust digital watermarking algorithm for ownership verification of digital images. In Computer Networks and Intelligent Computing: 5th International Conference on Information Processing, ICIP 2011, Bangalore, India, August 5-7, 2011. Proceedings (pp. 482-491). Springer Berlin Heidelberg.

[6] V. Goar, A. Sharma, N. S. Yadav, S. Chowdhury and Y.-C. Hu, "IOT-based smart mask protection against the waves of covid-19", Journal of Ambient Intelligence and Humanized Computing, 2022.

[7] Yavuz, E., & Telatar, Z. (2006, September). SVD adapted DCT domain DC subband image watermarking against watermark ambiguity. In International Workshop on Multimedia Content Representation, Classification and Security (pp. 66-73). Berlin, Heidelberg: Springer Berlin Heidelberg

[8] Shen, R. M., Fu, Y. G., & Lu, H. T. (2005). A novel image watermarking scheme based on support vector regression. Journal of Systems and Software, 78(1), 1-8.

[9] Huang, S., & Zhang, W. (2009, May). Digital watermarking based on neural network and image features. In 2009 Second International Conference on Information and Computing Science (Vol. 2, pp. 238-240). IEEE.

[10] Tang, G., & Liao, X. (2004). A neural network-based blind watermarking scheme for digital images. In Advances in Neural Networks-ISNN 2004: International Symposium on Neural Networks, Dalian, China, August 19-21, 2004, Proceedings, Part II 1 (pp. 645-650). Springer Berlin Heidelberg.

[11] R. Dash, T. N. Nguyen, K. Cengiz and A. Sharma, "FTSVR: Fine-tuned support vector regression model for stock predictions", Neural Computing and Applications, 2021, [online] Available: https://10.1007/s00521-021-05842-w.

[12] S. Samanta, A. Sarkar and A. Sharma, "Networking Technologies and challenges for green IOT applications in urban climate", Machine Learning and Data Science, pp. 169-184, 2022.

[13] Singh, P. K. (2022). Robust and imperceptible image watermarking technique based on SVD, DCT, BEMD, and PSO in wavelet domain. Multimedia Tools and Applications, 81(16), 22001-22026.

[14] A. K. Vashishtha, M. Sharma and A. Sharma, "Mechanism Incorporating Secure Mutual Validation and Key Spreading Organization in Intelligent Transport System," 2022 International Conference on Fourth Industrial

Revolution Based Technology and Practices (ICFIRTP), Uttarakhand, India, 2022, pp. 219-225, doi: 10.1109/ICFIRTP56122.2022.10059443.
[15] Saqib, M., & Naaz, S. (2017). Spatial and frequency domain digital image watermarking techniques for copyright protection. Int. J. Eng. Sci. Technol.(IJEST), 9(6), 691-699.
[16] Ko, L. T., Chen, J. E., Hsin, H. C., Shieh, Y. S., & Sung, T. Y. (2012). A unified algorithm for subband-based discrete cosine transform. Mathematical Problems in Engineering, 2012.
[17] Mehta, R., Rajpal, N., & Vishwakarma, V. P. (2015). Sub-band discrete cosine transform-based greyscale image watermarking using general regression neural network. International Journal of Signal and Imaging Systems Engineering, 8(6), 380-389.
[18] G. Sonowal, A. Sharma and L. Kharb, "Spear-phishing emails verification method based on verifiable secret sharing scheme", Journal of Information Assurance & Security, vol. 16, no. 3, pp. 117-124, 2021.
[19] Dixit, A., Agarwal, R. P., & Sharma, B. K. (2023, May). Hybridization of Discrete Cosine Transform and Principal Component Analysis to Achieve Digital Watermarking. In 2023 International Conference on Disruptive Technologies (ICDT) (pp. 527-530). IEEE.
[20] V. Goar, A. Sharma and D. Chahal, "Android Asset Packaging Tool based Forensics Security and Predictive Analysis", Journal of Information Assurance & Security, vol. 16, no. 3, pp. 124-131, 2021.
[21] Mehta, R. and Rajpal, N. (2013) 'General regression neural network based image watermarking using fractional DCT-II transform', Proceedings of the IEEE 2nd International Conference on Image Information Processing (ICIIP), 9–11 December, Shimla, Himachal Pradesh, India, pp.340–345.
[22] Singh, A. K., Dave, M., & Mohan, A. (2014). Hybrid technique for robust and imperceptible dual watermarking using error correcting codes for application in telemedicine. International Journal of Electronic Security and Digital Forensics, 6(4), 285-305.
[23] Begum, M., & Uddin, M. S. (2020). Digital image watermarking techniques: a review. Information, 11(2), 110.
[24] Latif, A., Naghsh-Nilchi, A. R., & Monadjemi, S. A. (2010). A parametric slant-Hadamard system for robust image watermarking. Journal of Circuits, Systems, and Computers, 19(02), 451-477.
[25] Sung, T. Y., Shieh, Y. S., & Hsin, H. C. (2010). An efficient VLSI linear array for DCT/IDCT using subband decomposition algorithm. Mathematical Problems in Engineering, 2010.
[26] Gafurov, A., Mukharamova, S., Saveliev, A., & Yermolaev, O. (2023). Advancing Agricultural Crop Recognition: The Application of LSTM Networks and Spatial Generalization in Satellite Data Analysis. Agriculture, 13(9), 1672.