UNIVERSITATY OF
BUCHAREST

FACULTY OF
MATHEMATICS AND
INFORMATICS

SPECIALIZATION INFORMATICS

Bachelor's thesis

IMAGE WATERMARKING THROUGH SPECTRAL ANALYSIS

Graduate
Dogaru Mihail Dănuț

Scientific coordinator
Conf. Dr. Rusu Cristian

Bucharest, June 2025

Rezumat

În această lucrare, sunt propuse o varietate de algoritmi de inserare de informație ascunsă în domeniul frecvenței pentru imagini binare, reprezentând drepturile de autor pentru imagini gazdă alb-negru. Obiectivul principal este de a analiza comportamentul și robustețea a diferite metode cuprizând Transformata Fourier, Spread Spectrum, Transformata Wavelet, Wavelet-SVD, după o varietate de atacuri asupra imaginii. Atacurile variază între Zgomot Gaussian Aditiv, Compresie JPEG, Filtrare, Egalizarea Histogramei si Transformari Geometrice.

Abstract

In this paper, multiple watermarking schemes in the frequency domain are proposed for embedding binary images, representing ownership over a grayscale host image. The main goal is to analyze the behavior and robustness over various image attacks of different methods involving Fourier Transform, Spread Spectrum, Wavelet Transform, and Wavelet-SVD. The attacks range from Additive Gaussian Noise, JPEG Compression, Filtering, Histogram Equalization, and Geometric Transformations.

Contents

Chapter 1

Introduction

1.1 Watermarking

When we talk about digital multimedia, we refer to electronic signals such as songs, videos, images, and we call them all generically *content*. When we talk about a particular material, we name it *Work* or *Cover Work* and the representation or transmission channel is named *media*. For example, music is such a *content*, the song "Billie Jean" of Michael Jackson is a *Work*, stored on a CD, which is its *media*.

A similar research area within the framework of Information Security is *Steganography*, many techniques being shared for both purposes, having basically the same generic system. While Steganography aims to *"undetectably alter a Work to embed a secret message"*, Watermarking deals with *"imperceptibly altering a Work to embed a message about that Work"* [5], the key difference being that the existence of the watermark is not a secret.

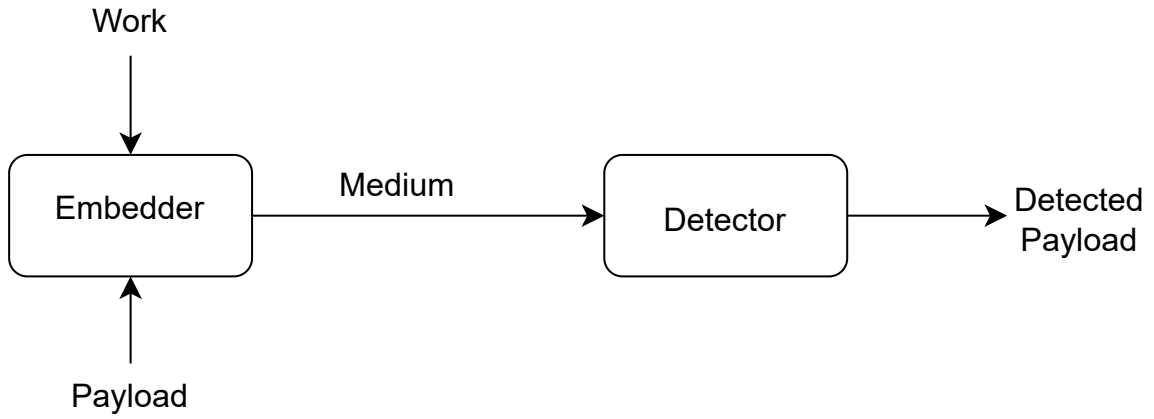

Figure 1.1: Generic Watermark/Steganography System

1.2 Motivation

As multimedia content has grown over the past decades, data security and intellectual property issues have become critical. Watermarking ensures traceability in areas like medical records, digital assets, and authentication. Due to the seriousness of these applications, robust systems resistant to attacks and transmission degradation are essential.

Many systems fall into two classes based on the exploited domain: space (or time) domain and frequency domain. The frequency domain offers better imperceptibility and robustness but involves greater complexity and lower data payload capacity.

Usually, the receiver lacks access to the original work, creating the need for blind systems that rely on techniques like quantization for watermark detection.

1.3 Personal Contribution

This thesis adapts and analyzes four watermarking systems, each exploiting properties of digital grayscale images (2D signals) and binary watermarks, to determine which performs best in different contexts. The analysis covers maximum payload capacity, extraction error, and imperceptibility to the Human Visual System (HVS). Finally, an MLP network is used to identify which watermarking method was applied. The implementations are available in a public repository online[1].

1.4 Related Works

In the digital medium, special attention is paid to the geometric distortions provided by any editing software, and so several methods [2, 9, 10, 14] have been proposed, based on various properties of the frequency domain, to achieve higher robustness. A common and effective technique is to combine space domain and frequency domain alterations for a better imperceptibility, a well known mean being the Discrete Wavelet Transform [7, 17]. Notably, hybrid schemes that incorporate Singular Value Decomposition have shown promising results [1, 3, 18]

1.5 Structure

In Chapter 2, we detail the steps of four watermarking methods, covering both embedding and extraction processes, while highlighting key observations. Chapter 3 provides an analysis of the methods' performance under various attacks through quantitative testing. In Chapter 4, we introduce a Multi-Layer Perceptron (MLP) to develop a general-purpose detector.

[1] Full Source Code

Chapter 2

Metodology

2.1 Preprocessing

The preprocessing steps begin by converting both the work and the watermark images to grayscale. Next, the watermark is resized to the proper dimensions to fit the embedding scheme. After resizing, the watermark is thresholded to obtain a binary image, which is then normalized from the range $[0, 255]$ to $[0, 1]$. Finally, the binary watermark is scrambled using a chosen key that is smaller than its side length.

2.2 Watermark Scrambling

For securing confidentiality, in the case of watermarking, encryption is crucial, so that only authorized entities are able to access private information, a malicious adversary being unable to get, alter, or remove the data.

In all the following schemes, the watermark is first scrambled by *Arnold's Cat Map* [13], a chaotic, periodic permutation, defined on the quotient space $\mathbb{T} := \mathbb{R}^2/\mathbb{Z}^2$, with the goal of hiding the secret image as a noise pattern, hiding it's original structure. For an image of size $N \times N$, we define:

$$\Gamma : \mathbb{T}^2 \times \mathbb{N} \to \mathbb{T}^2, \quad \Gamma(\begin{bmatrix} x \\ y \end{bmatrix}, k) = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}^k \begin{bmatrix} x \\ y \end{bmatrix} (mod\ N) \tag{2.1}$$

where $\begin{bmatrix} x \\ y \end{bmatrix}$ corresponds to the pixel at line x and column y and the scrambling key k.

As shown in Figure 2.2, there is no direct link between the side length of the image and the period of the image, which is currently the subject of multiple studies.

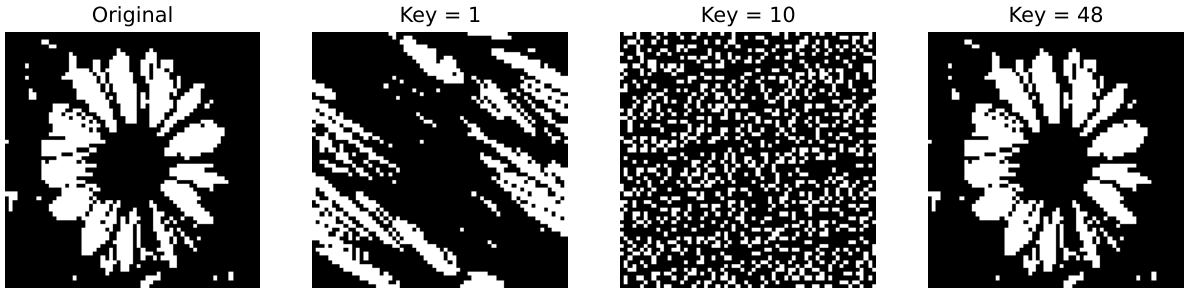

Figure 2.1: Cat Map Period for 64x64 Image

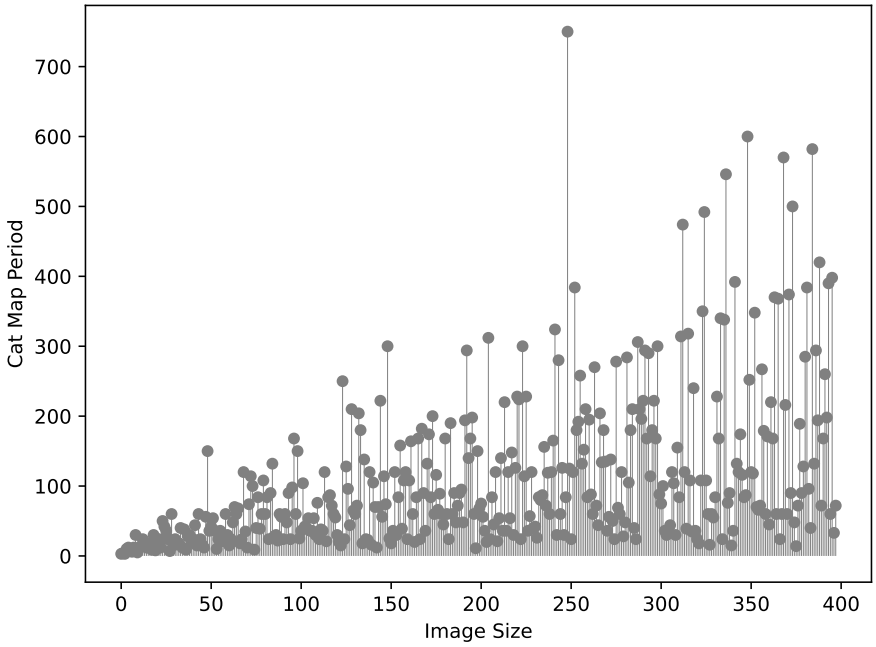

Figure 2.2: Distribution of period lengths for Arnold's Cat Map across square image sizes ($N \times N$). Each dot represents a distinct period for a given size, revealing increasingly complex and varied behavior as image size grows.

2.3 Basic Fast Fourier Transform Embedding

2.3.1 Embedding

The first system employs a simple, yet effective embedding scheme. First, we compute the 2D Fast Fourier Transform[1] of the Work H, resulting in its spectrum F. The encrypted watermark w_m is multiplied by a factor of e^{α}, where α is the embedding strength, giving control over the trade-off between robustness and visibility of the watermark.

We are left with the choice of adding the watermark in low-frequency or high-frequency regions of the spectrum[2], each offering advantages depending on the application.Both

[1] 2D Fast Fourier Transform

[2] The *fftshift* in NumPy moves the low frequency components to the corners, shifting the zero-frequency to the center: np.fft.fftshift

situations will be explored.

Algorithm 1 Basic FFT Embedding

Require: Host image H, Watermark W, Embedding strength α, Flag `fftshift`

$F \leftarrow \texttt{FFT2}(H)$
if `fftshift` is True **then**
    $F \leftarrow \texttt{FFTShift}(F)$
end if
Pad W to the center of F's shape: $W \leftarrow \texttt{PadToCenter}(W, \text{shape}(F))$
Compute embedded spectrum: $F' \leftarrow F + e^{\alpha} \cdot W$
if `fftshift` is True **then**
    $F' \leftarrow \texttt{IFFTShift}(F')$
end if
Inverse FFT: $\hat{H} \leftarrow \texttt{RealPart}(\texttt{IFFT2}(F'))$
Clamp values to $[0, 255]$: $\hat{H} \leftarrow \texttt{Clip}(\hat{H}, 0, 255)$
return $\hat{H}$

2.3.2 Extraction

The detection procedure is less sophisticated, needing only a cropped portion of the size of the original watermark from the corresponding section of the spectrum, over which we can apply a threshold to regain a binary image.

2.3.3 Detection

This simple method allows a detection test before extraction, if we have access to the original encrypted watermark. Let c_0 be our Work and w the watermark, therefore, the watermarked image will be expressed as $c = c_0 + e^{\alpha} w$. We define the correlation coefficient[3] as: $z_{cc}(v, u) = \frac{v \cdot u}{\sqrt{(v \cdot v)(u \cdot u)}}$. If $c = c_0 + e^{\alpha} w + n$, where n represents Gaussian noise and ignoring the constants, to test a possible watermark x, the computation becomes:

$$\begin{aligned} z_{cc}(c, x) &= x \cdot (c_0 + w + n) \\ &= x \cdot c_0 + x \cdot w + x \cdot n \end{aligned} \tag{2.2}$$

By choosing x as close as possible to a noise pattern with the scrambling, all terms cancel out by orthogonality, leaving just $x \cdot w$, so z_{cc} ignores the Work and any Gaussian noise and gives the correlation between the candidate watermark and the actual watermark.

[3] Correlation Coefficient

2.3.4 Results

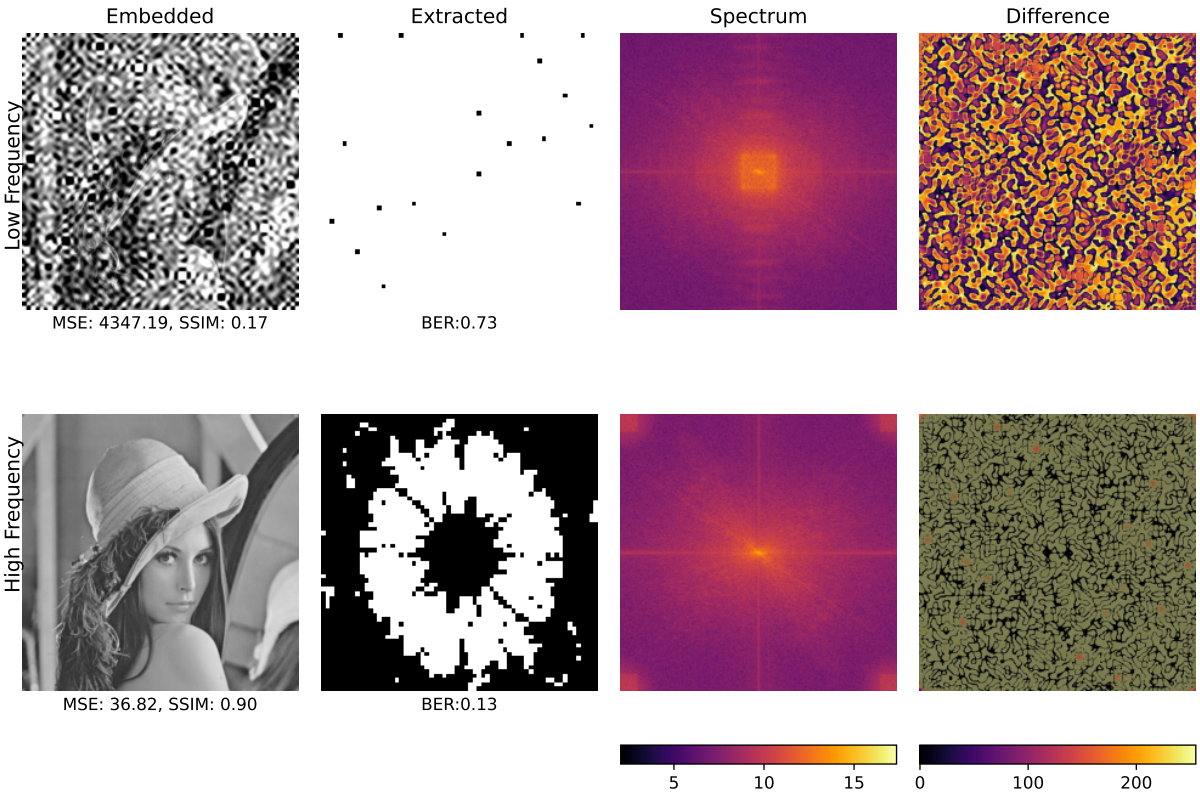

Figure 2.3: Comparing Shifted and Unshifted Versions with Best Embedding Strength

2.3.5 Observations

For an image of $N \times M$ pixels, the maximum payload this method can embed is technically $N \times M$ bits, but the quality will be reduced due to the action on the low frequency components.

Usually when plotting the 2D spectrum we do it in logarithmic scale due to high variations of magnitudes, a linear scale would make the small values almost invisible. So, having α on an exponential scale will make the tuning easier.

As shown in Figures 2.4 and 2.5 the choice of the key does affect the value of $z_{cc}(c, x)$, so finding a suitable threshold to decide the presence of x is no easy task, but before embedding we can iteratively find the α that gives the best correlation.

This method allows multiple watermarks to be added to an image, but if it is done in the same section of the spectrum, the correlations will be reduced for each one of them.

Due to the low perception of the HVS for high frequencies, the unshifted version performs better, leaving only subtle artifacts, the other one needing a higher embedding exponent to reach a peak in correlation, but will be clipped away and will not serve a practical purpose, also applying a deep frying effect.

For a blind detector, the high frequency methods seems the better choice, but the low frequency approach can be viable for an informed detector, granting resistance to another range of attacks, so it's still worth mentioning.

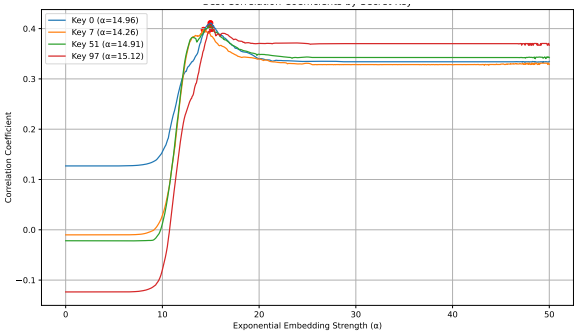

Figure 2.4: Best Embedding Strength at Low Frequency for Multiple Keys

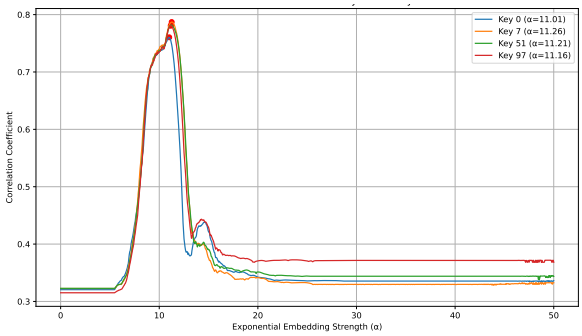

Figure 2.5: Best Embedding Strength at High Frequency for Multiple Keys

2.4 Spread Spectrum Fourier Transform Embedding

This method is an adaptation of the work of Cox et al.[4], proposing a spread spectrum watermarking scheme. The key differences are the use of quantization in the embedding process, followed, of course, by the associated detector, and also the use of FFT instead of DCT[4].

2.4.1 Embedding

Many attacks, processing, and distortions have an impact on the "insignificant regions of the image (or spectrum)", an immediate decision can be adding the watermark on the most significant components that can not be easily altered or removed.

To tackle this task, we sort the indices of the FFT components of the Work descendingly by their magnitude and quantize each component by a bit from the flattened watermark, treating it as a 1D signal.

2.4.2 Extraction

In the embedder, the selected coefficients are adjusted to fall in the offset 0.25 or 0.75, multiplied by a factor of α. In the extractor, the chosen components are divided by α, if the fraction is greater than 0.5, it is interpreted as a 1, else a 0.

[4]Discrete Cosine Transform

Algorithm 2 Spread Spectrum FFT Embedding with Quantization

Require: Host image H, Watermark W, Secret key k, Embedding strength α

Flatten watermark: $w \leftarrow \texttt{Flatten}(W)$
Compute FFT: $F \leftarrow \texttt{FFT2}(H)$
Magnitude and phase: $M \leftarrow |F|$, $\Phi \leftarrow \angle F$
Flatten: $m \leftarrow \texttt{Flatten}(M)$, $\phi \leftarrow \texttt{Flatten}(\Phi)$
Sort indices by descending magnitude: $I \leftarrow \texttt{Argsort}(-m)$
Select top-N indices: $I_s \leftarrow I[0{:}\texttt{len}(w)]$
for $i = 0$ to $\texttt{len}(w) - 1$ **do**
    $idx \leftarrow I_s[i], \quad b \leftarrow w[i]$
    $q \leftarrow \lfloor m[idx]/\alpha \rfloor$
    if $b = 0$ **then**
        $m[idx] \leftarrow \alpha \cdot (q + 0.25)$
    else
        $m[idx] \leftarrow \alpha \cdot (q + 0.75)$
    end if
end for
Reconstruct modified spectrum: $F' \leftarrow m \cdot e^{j \cdot \phi}$, reshape to original FFT shape
Inverse FFT: $\hat{H} \leftarrow \texttt{RealPart}(\texttt{IFFT2}(F'))$
Clamp values: $\hat{H} \leftarrow \texttt{Clip}(\hat{H}, 0, 255)$
return $\hat{H}$

2.4.3 Results

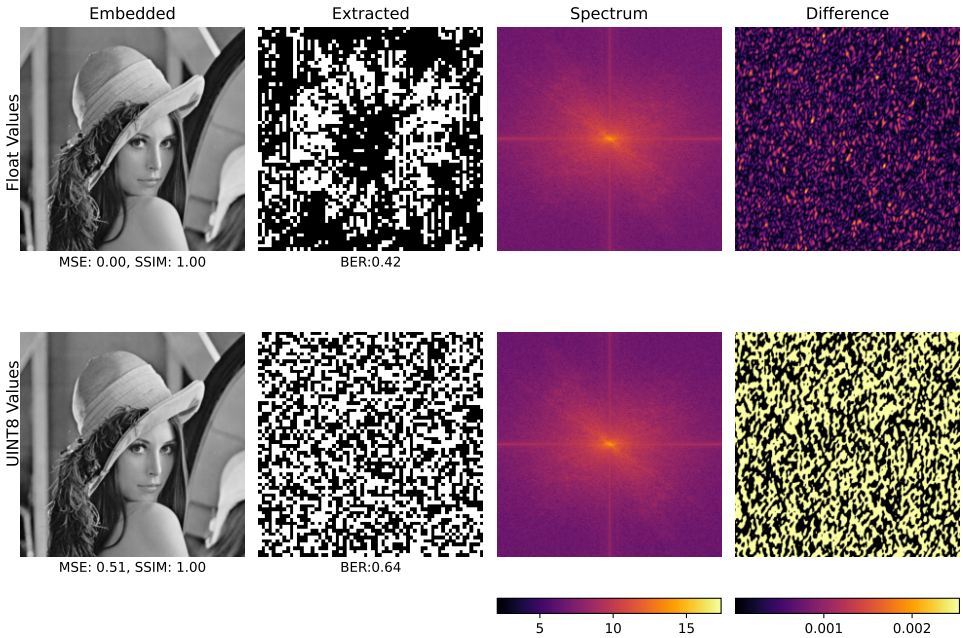

Figure 2.6: Comparing Results for Values Stored as Float or UINT8

2.4.4 Observations

For an image of $N \times M$ pixels, the maximum payload that can be embedded is $N \times M$ bits, exactly one bit from the watermark for each FFT component.

A quantization scheme can perform poorly applied on the spectrum, slight modifications in the FFT components can result in fractional values in the space domain that will be lost on clipping and casting to *uint8*, the only option is saving the values as floats, which makes the digital medium a poor choice. The approach of taking the first components by their magnitudes might be problematic. After embedding, there is no guarantee that the components we attempt to extract remain the same.

The changes for the float version are so slight that the error is virtually rounded to 0.

As seen in Figure 2.6, the difference heatmap between the original Cover Work and the embedded one presents a higher disorder, which means a higher entropy.

Embedding strength is additionally needed for extraction, which means that we need additional information for extraction.

2.5 Basic Wavelet Embedding

This embedding scheme is an implementation of the work of Raval and Rege [11], proposing a 2D discrete wavelet transform[5] approach.

The main idea is to combine modifications in the low frequency and high frequency sub-bands of the DWT taking advantage of both of their properties against attacks and robustness.

2.5.1 Embedding

Firstly, we perform the 2nd level DWT transform, resulting in 7 sub-bands, 3 from the first level transform, and 4 from applying another 1 level transform on the HH sub-band from the one before, as shown in Figure 2.7.

And secondly, we add the watermark to the highest and lowest sub-bands.

In this experiment, the Dauberchies 1 wavelet [6] was chosen as the mother wavelet.

2.5.2 Extraction

In the original paper, the system was informed, having access to the original Cover Work, the task of extraction becoming trivial, directly from the sub-band difference of the two images.

[5] 2D Discrete Wavelet Transform
[6] Daubechi Wavelet

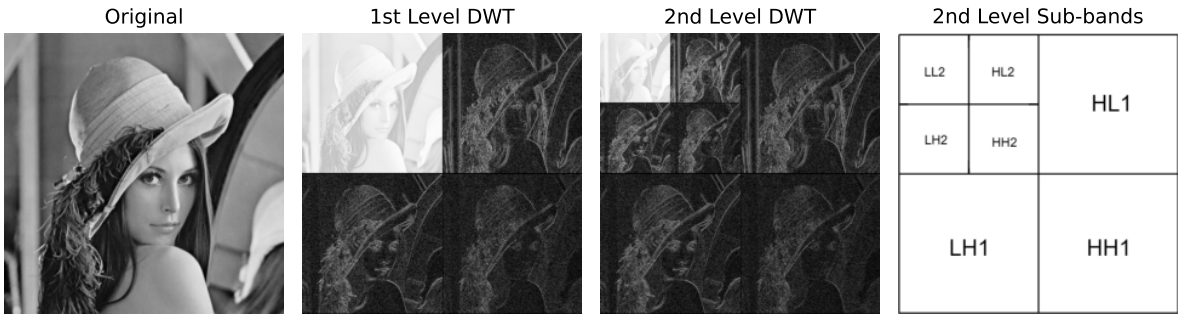

Figure 2.7: 2nd Level DWT Structure

Algorithm 3 DWT-based Embedding

Require: Host image H, Watermark W, Secret key k, Embedding strength α

1: Decompose H with 2D wavelet transform: $\{C_i\} \leftarrow \texttt{DWT2}(H, \text{level} = 2)$
2: Pad W to center of HH band shape: $W_{HH} \leftarrow \texttt{PadToCenter}(W, \text{shape}(C_L^{HH}))$
3: Pad W to center of LL band shape: $W_{LL} \leftarrow \texttt{PadToCenter}(W, \text{shape}(C_0))$
4: Modify HH band: $C_L^{HH} \leftarrow C_L^{HH} + \alpha \cdot W_{HH}$
5: Modify LL band: $C_0 \leftarrow C_0 + \alpha \cdot W_{LL}$
6: Update coefficients with modified bands
7: Reconstruct image: $\hat{H} \leftarrow \texttt{IDWT2}(\{C_i\})$
8: **return** $\hat{H}$

By assuming that the system does not have access to the original, we can use the approach from 2.3.2 for the HH sub-band, the LL containing more energy making it too difficult to filter. In addition, we can use the median as the threshold.

2.5.3 Results

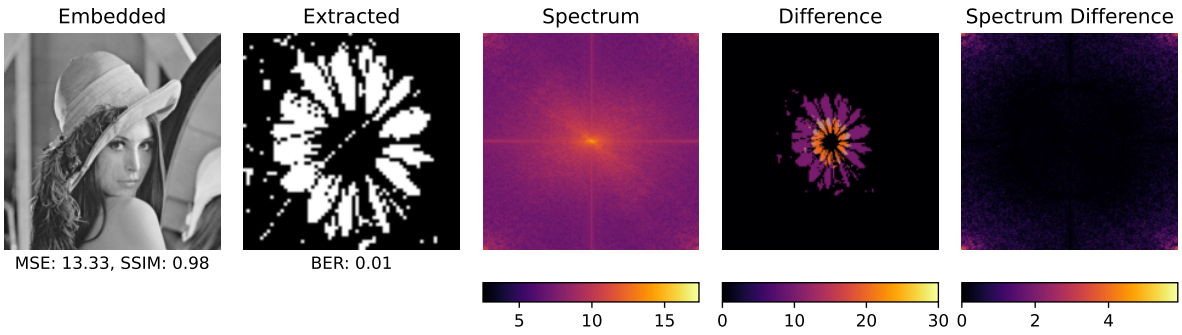

Figure 2.8: Results and Differences in Time and Frequency Domains

2.5.4 Observations

The maximum payload possible for a Cover Work of shape $N \times M$ pixels is $\frac{N}{4} \times \frac{M}{4}$ bits, which is the shape of the LL sub-band that restricts us.

As shown in Figure 2.8, the addition in the DWT component before reconstruction has a scattering effect of the watermark over the spectrum, creating small artifacts, stronger in the high frequency sections. Unlike the previous two methods, this scheme affects time-domain regions of the Work, producing a scaling effect on the watermark.

In Figure 2.8 the watermark is not scrambled to better showcase the effect of altering the sub-bands, we can notice 2 overlapping flowers, the inner one coming from the HH sub-band, and the outer one from the LL sub-band. The scaling effect is produced by the scalings needed to reconstruct the image with the inverse transform, each sub-band being in the end rescaled to the original size of the Work.

2.6 DWT-SVD Embedding

This method is an adaptation of the work of Tsai et al.[15], proposing an embedding scheme in the Wavelet domain, based on the singular value decomposition (SVD) [12].

The original paper uses a support vector regression (SVR) extractor. This implementation adjusts the embedding with a quantization step for an easier extraction.

2.6.1 Embedding

First, we perform 1st level DWT on the Cover Work and extract the LL sub-band.

Secondly, we break the LL into non-overalapping square patches of the longest side length to fit the whole watermark and spread the modification over as much pixels as possible to increase entropy.

Thirdly, we apply SVD on each block and quantize the first singular value with the corresponding bit from the watermark.

Lastly, we reconstruct each patch and apply IDWT to obtain the watermarked Work.

Algorithm 4 DWT-SVD Embedding with Quantization

Require: Square host image H, square binary watermark W, secret key k, embedding strength α

Apply 2D wavelet transform to H to obtain LL subband
Compute patch size: $p \leftarrow \text{size}(LL)/\text{size}(W)$
Flatten watermark: $W_f \leftarrow \text{flatten}(W)$
Break LL into patches of size $p \times p$
for each bit b_i in W_f **do**
    Perform SVD on patch P_i: $[U, S, V] \leftarrow \text{SVD}(P_i)$
    Modify singular value: $S_0 \leftarrow \alpha \cdot \text{round}\left(\frac{S_0 - (\alpha/2) \cdot b_i}{\alpha}\right) + (\alpha/2) \cdot b_i$
    Reconstruct patch: $\hat{P}_i \leftarrow U \cdot \text{diag}(S) \cdot V$
    Store $\hat{P}_i$ in list of embedded patches
end for
if remaining patches exist **then**
    Append unmodified patches to embedded patches list
end if
Reconstruct modified LL from embedded patches
Replace original LL with modified LL in wavelet coefficients
Apply inverse wavelet transform to obtain $\hat{H}$
Clamp values: $\hat{H} \leftarrow \texttt{Clip}(\hat{H}, 0, 255)$
return $\hat{H}$

2.6.2 Extraction

The extraction procedure is similar. Apply 1st level DWT and break the LL sub-band to patches. Apply SVD on each block and take the first singular value to determine which lattice was used, bit 0: $[0, 0.25] \cup [0.75, 1]$ and bit 1: $(0.25, 0.75)$.

2.6.3 Observations

For an image of N^2 pixels, the maximum payload that can be embedded depends on the patch size chosen, but for a patch size of 1 i.e. 1 bit per pixel, we can embed a maximum of $N^2/4$ bits, due to the LL sub-band's size.

The LL sub-band contains most of the energy and structure of the image, making it less likely to be affected by attacks. Unlike higher frequency bands that retain details and edges, easily perturbed by compression or noise.

SVD is a numerically stable method that captures structural information of the image, singular values being strongly related to the luminance energy. Singular values come in descending order, the first one packing the most information about the image, as shown in Figure2.9.

By embedding the watermark in the LL sub-band instead of the whole image, we obtain a smaller conditioning number[7] by a few orders of magnitude, making it more

[7] Condition number

stable.

We can notice that for a small patch size, visible artifacts start to appear, and the error increases. In addition, for a watermark of a size that does not divide the Cover Work's size, the altered pixel will start from the top of the image, meaning a lower entropy.

2.6.4 Results

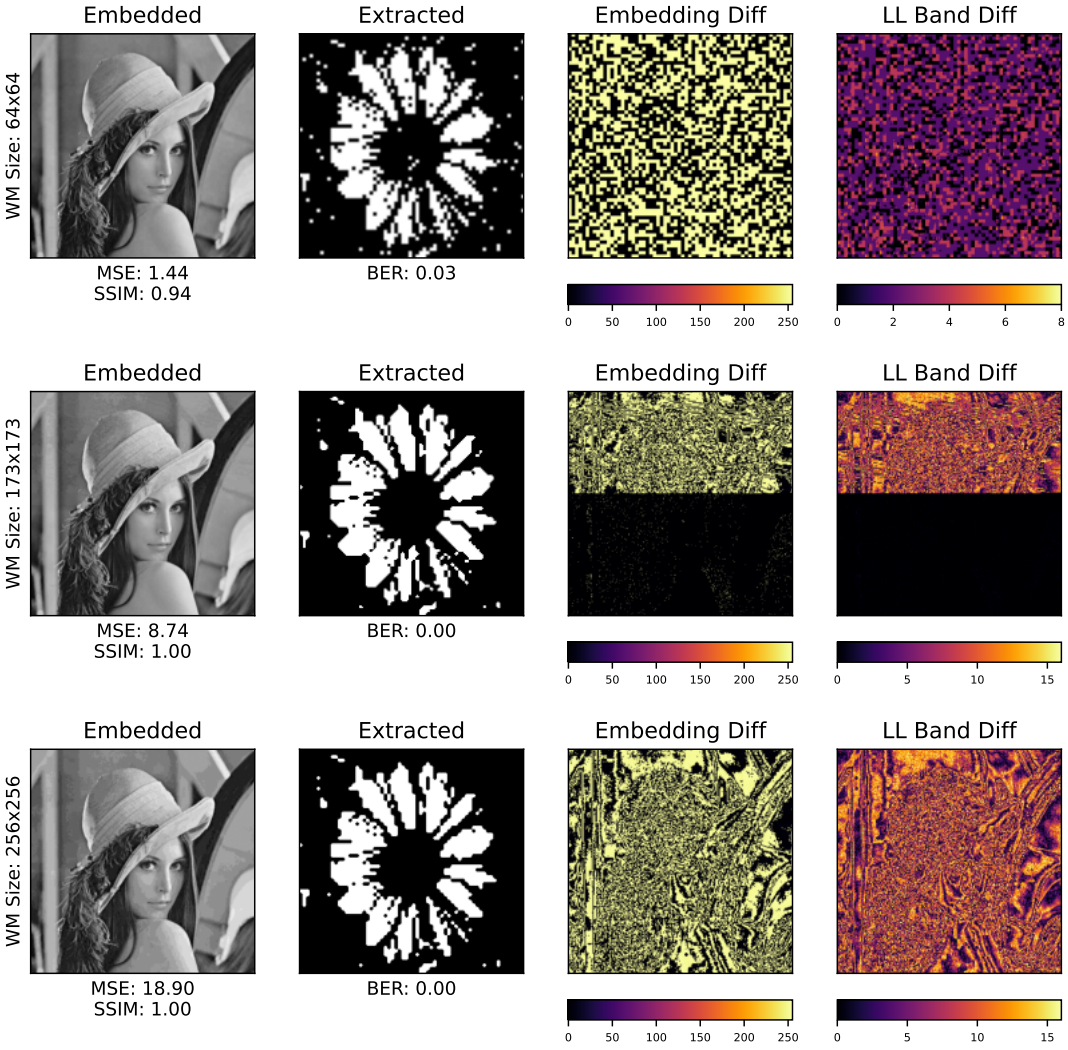

Figure 2.9: Comparing Results for Multimple Watermark sizes over Work of shape 512x512

Chapter 3

Experiments

In this chapter, we will observe the behavior of each embedding scheme under multiple common noise attacks, filtering, geometric distorsions, and compression.

3.1 Documented Attacks

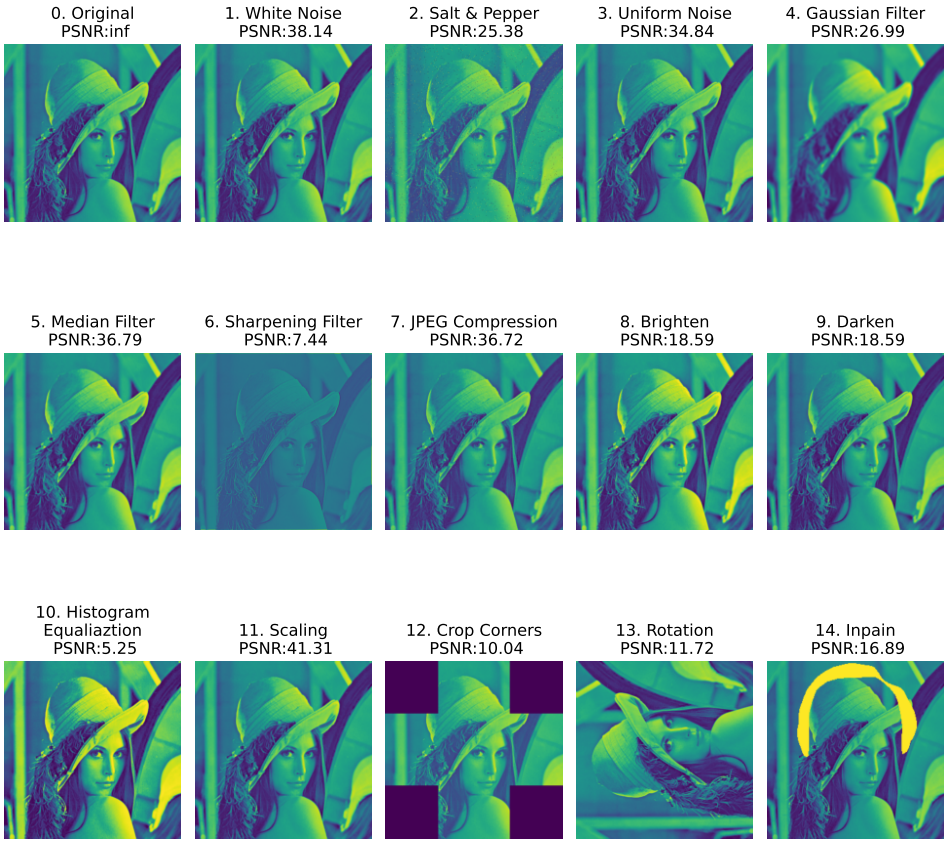

Figure 3.1: Documented attacks

3.2 Watermarks after Attacks

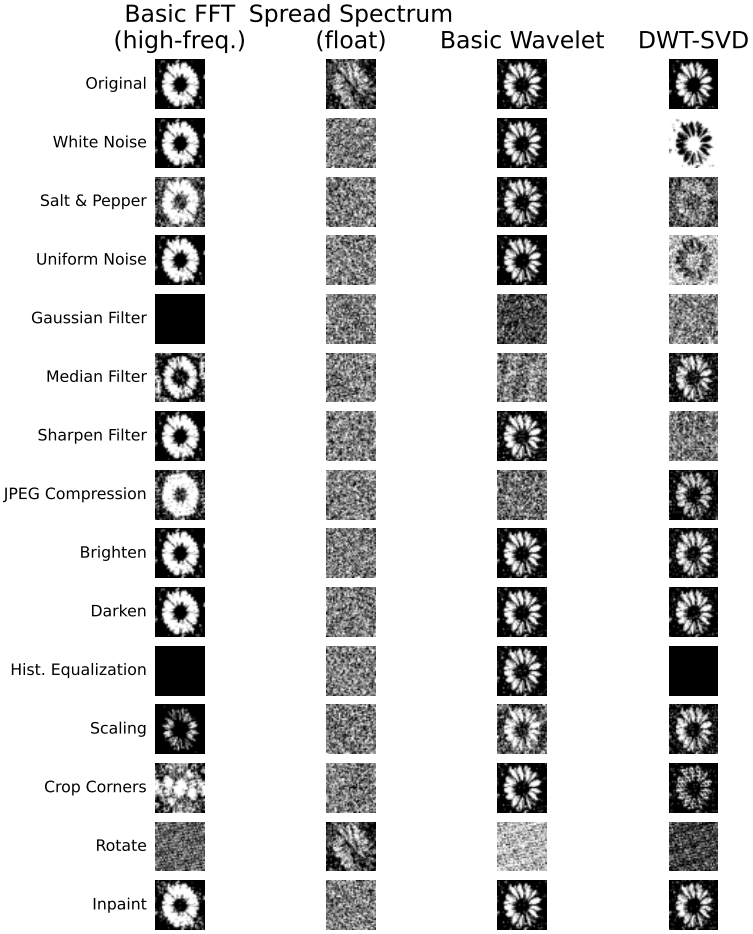

Figure 3.2: Effects of the Attacks over All Embeddings

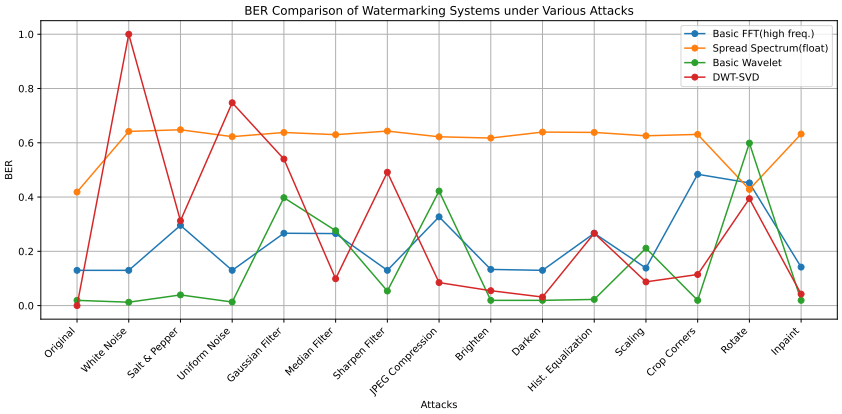

Figure 3.3: BER Analysis of Watermarking Systems Under 15 Attacks

3.3 Attack Results Interpretation

We can notice for the Basic FFT resisting to the JPEG Compression due to the exponential embedding strength, but being killed off by the Gaussian Filter that discards that region from the spectrum entirely.

The Spread Spectrum method, unsurprisingly, has poor performance, but for the Rotation attack the answer is more subtle. While rotating an image, the spectrum components also rotate, but do not change energy and by sorting them we get the same values.

For the other methods, the Rotation attack succeeds due to the scrambling map. We get a rotated version of the scrambling and we try to unscramble as if it was on the correct position, resulting in a wrong pattern.

The Basic Wavelet Method comes in short for Filters in high frequencies, where the only solution could be watermarking in low frequencies, and fails for JPEG Compression, which could be adapted with a higher embedding strength. But it seems undisturbed by spatial perturbations.

DWT-SVD approach seems easily perturbed by spatial perturbations such as Salt & Pepper and Uniform Noise, those types of attacks ruining the structure of the blocks. Also, the Histogram Equalization redistributes pixels intensively, destroying the structure.

Interestingly, for the White Noise and Uniform Noise attacks, all the maximum singular values are pushed into the wrong quantization bins by the perturbations.

No embedding scheme performs well in all the possible situations, so the choice of embedding is strongly relied on the expected handling of the medium.

3.4 Embeddings Summary

Table 3.1: Comparison of Embedding Systems

	Basic FFT	**Spread Spectrum (float)**	**Basic Wavelet**	**DWT-SVD**
Average MSE	166.23	1.36×10^{-6}	12.95	2.45
Average SSIM	0.78	0.99	0.97	0.99
Average BER	0.46	0.49	0.13	0.10
Max Payload	$N \times M$ bits	$N \times M$ bits	$\frac{N}{4} \times \frac{M}{4}$ bits	$\frac{N^2}{4}$ bits
Detection	Via thresholding	Via spectrum quantization	Via median thresholding	Via SVD quantization
Domain	Frequency (FFT)	Frequency (FFT)	Wavelet (DWT)	Wavelet (DWT)+SVD
Robustness	Moderate (depends on frequency band)	Very Low (in uint8 format)	High (mostly in HH)	high (robust LL + stable SVD)
Artifacts	Visible if α too large; high-freq preferred	Very low if float is preserved; can fail with uint8	Slight artifacts from watermark scaling	Depends on patch size; small patches cause visible distortion
Artifacts Localization	Global (periodic patterns)	Global	Localized (edge distortions)	Localized (patches)
Entropy Effect	High in time; Low in frequency	High in time; High in frequency	Low in time; High in frequency	High in time, depending on image sizes; Very high in frequency
Implementation Complexity	Low	Moderate	Moderate	High

Chapter 4

MLP Classifier on Spectral Features

4.1 Motivation

The task of watermark detection becomes significantly more challenging when no information about the embedding scheme is available. In such cases, a general-purpose classifier is proposed to infer the presence or type of watermark based solely on frequency-domain features. Since artifacts introduced by embedding are often imperceptible in the spatial domain and difficult for the HVS to detect, analyzing the frequency spectrum offers a more objective and quantifiable signal for classification.

4.2 Data Collection

This project uses a custom dataset. The Cover Works consists of a subset of 10000 images from ImageNet[6], grayscaled and reshaped to 256×256. The watermarks are 32×32 images from cifar-10[8] dataset, binarized with a median threshold. Each Work is embeded with all 4 systems with a random watermark, resulting in 40000 images containing secret information.

4.3 Feature Extraction

The input data consists of images, of which we compute the 2D FFT, extract the log-magnitudes, flatten, and normalize with the z-score. Since each point in the spectrum corresponds to a specific frequency component, that can be treated independently, a Multi-Layer Perceptron(MLP) is an appropriate approach for classification.

4.4 Network Architecture

The network consists of 3 fully connected layers:

- An input layer, with an entry for each magnitude component, namely 256×256, maping into 128×128 perceptrons
- A hidden layer, that reduces dimensionality from 128×128 to 128
- An output layer, that maps the 128 to the desired number of classes, in our case 4.

The non-linear activation function ReLU is applied between all layers to introduce model capacity and enable learning of complex patterns.

4.5 Training Setup

The dataset was split into train, validation, and test sets. The model used cross-entropy loss to measure the difference between predicted probabilities and true labels, helping it improve accuracy.

4.6 Results and Limitations

The trained MLP model performed well on two of the four classes, likely due to stronger artifacts in their frequency spectrum. These classes were classified with near-perfect accuracy.

However, the model failed to generalize to the other two classes, often predicting one dominant label regardless of the true class (see Figure 4.1). This points to limited class separability or insufficient model capacity, only achieving at most 75% accuracy.

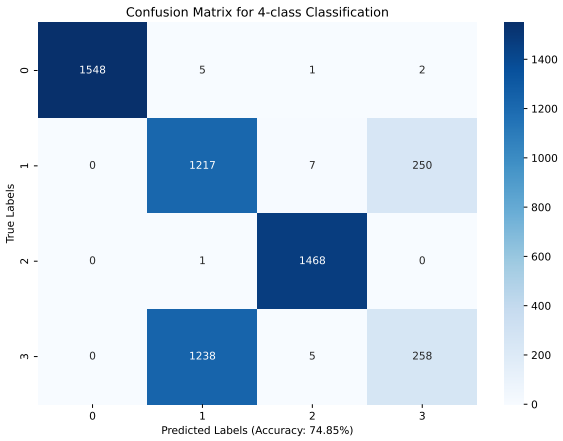

Figure 4.1: Confusion Matrix for Methods Classification

Adding spatial features during experimentation did not improve results, suggesting the issue lies deeper in the data representation or model architecture.

Chapter 5

Summary

5.1 Conclusion

This thesis presents an adaptation of several blind watermarking schemes that operate in the frequency domain. The strengths and weaknesses of each method have been evaluated across various scenarios, and a Multi-Layer Perceptron (MLP) has been proposed to provide a generalized detection approach.

We conclude that there is no universally optimal embedding and detection method; the appropriate choice depends on the expected transformations or distortions that the medium may impose on the watermarked image.

5.2 Future Work

Future improvements may include extending the methods for color images; for now, one can use any of the above methods on a color channel. An intriguing method can be found in the work of Wang et al.[16], proposing a Quaternion Fourier Transform, which has also been implemented as part of this work[1].

Incorporating concepts from Information Theory, such as Dirty Paper Coding, allowing the sender to take advantage of possible interferences, could contribute to more effective and robust embedding strategies.

Additionally, improving the neural network architecture, such as optimizing the MLP or exploring deeper or convolutional models, may lead to better performance and generalization for classification.

[1] Quaternion Fourier Transform for Watermarking Implementation

Bibliography

[1] Paul Bao and Xiaohu Ma. "Image adaptive watermarking using wavelet domain singular value decomposition." In: *IEEE transactions on circuits and systems for video technology* 15.1 (2005), pp. 96–102.

[2] Mauro Barni, Franco Bartolini, Vito Cappellini, and Alessandro Piva. "A DCT-domain system for robust image watermarking." In: *Signal processing* 66.3 (1998), pp. 357–372.

[3] Mahbuba Begum and Mohammad Shorif Uddin. "Implementation of secured and robust DFT-based image watermark through hybridization with decomposition algorithm." In: *SN Computer Science* 2.3 (2021), p. 221.

[4] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. "Secure spread spectrum watermarking for multimedia." In: *IEEE transactions on image processing* 6.12 (1997), pp. 1673–1687.

[5] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. 2nd. Burlington, MA: Morgan Kaufmann, 2007.

[6] Addison Howard, Eunbyung Park, and Wendy Kan. *ImageNet Object Localization Challenge*. https://kaggle.com/competitions/imagenet-object-localization-challenge. Kaggle. 2018.

[7] H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura. "A digital watermark based on the wavelet transform and its robustness on image compression." In: *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*. Vol. 2. 1998, 391–395 vol.2. DOI: 10.1109/ICIP.1998.723388.

[8] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009. URL: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[9] Vinicius Licks and R Hordan. "On digital image watermarking robust to geometric transformations." In: *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*. Vol. 3. IEEE. 2000, pp. 690–693.

[10] Benjamin Mathon, Patrick Bas, François Cayre, and Benoît Macq. "Comparison of secure spread-spectrum modulations applied to still image watermarking." In: *Annals of Telecommunications-annales des télécommunications* 64.11 (2009), p. 801.

[11] M.S. Raval and P.P. Rege. "Discrete wavelet transform based multiple watermarking scheme." In: *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*. Vol. 3. 2003, 935–938 Vol.3. DOI: 10.1109/TENCON.2003.1273384.

[12] Rowayda A Sadek. "SVD based image processing applications: state of the art, contributions and research challenges." In: *arXiv preprint arXiv:1211.7102* (2012).

[13] Fredrik Svanström. *Properties of a generalized Arnold's discrete cat map*. 2014.

[14] R Tay and JP Havlicek. "Image watermarking using wavelets." In: *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.* Vol. 3. IEEE. 2002, pp. III–III.

[15] Hung-Hsu Tsai, Yu-Jie Jhuang, and Yen-Shou Lai. "An SVD-based image watermarking in wavelet domain using SVR and PSO." In: *Applied Soft Computing* 12.8 (2012), pp. 2442–2453.

[16] Xiang-yang Wang, Chun-peng Wang, Hong-ying Yang, and Pan-pan Niu. "A robust blind color image watermarking in quaternion Fourier transform domain." In: *Journal of Systems and Software* 86.2 (2013), pp. 255–277.

[17] Yiwei Wang, John F Doherty, and Robert E Van Dyck. "A wavelet-based watermarking algorithm for ownership verification of digital images." In: *IEEE transactions on image processing* 11.2 (2002), pp. 77–88.

[18] Fauzia Yasmeen and Mohammad Shorif Uddin. "An efficient watermarking approach based on LL and HH edges of DWT–SVD." In: *SN Computer Science* 2.2 (2021), p. 82.