**DOKUZ EYLÜL UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# DIGITAL IMAGE WATERMARKING USING FRACTIONAL FOURIER TRANSFORM

**by**

**Olcay DUMAN**

**July, 2024**

**İZMİR**

# DIGITAL IMAGE WATERMARKING USING FRACTIONAL FOURIER TRANSFORM

**A Thesis Submitted to the**
**Graduate School of Natural and Applied Sciences of Dokuz Eylül University**
**In Partial Fulfillment of the Requirements for Master of Science in**
**Department of Electrical and Electronics Engineering**

**by**
**Olcay DUMAN**

**July, 2024**

**İZMİR**

**THESIS EXAMINATION RESULT FORM**

We have read the thesis entitled **"DIGITAL IMAGE WATERMARKING USING FRACTIONAL FOURIER TRANSFORM"** completed by OLCAY DUMAN under supervision of **PROF.DR. OLCAY AKAY** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof.Dr. Olcay AKAY

Supervisor

Doç.Dr. Yavuz ŞENOL

Jury Member

Doç.Dr. Burhan GÜLBAHAR

Jury Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

## ACKNOWLEDGMENT

First, I would like to state my sincere thanks to my advisor Prof. Dr. Olcay AKAY for his ongoing help, patience, vast expertise and support. His guidance and mentorship have been priceless during this thesis study. I must express his encouragement. His perspective feedback and recommendations significantly enhanced my thesis. Then, I want to say a big thank to my parents for always supporting me in my studies. They believed in me and kept pushing me to do my best. I'm also grateful to my family (my wife and my daughter Ece) who were there for me when I needed someone to talk to, gave me good advice, and helped me relax when things got tough. Their belief made this whole experience a lot easier.

Olcay DUMAN

## DIGITAL IMAGE WATERMARKING USING FRACTIONAL FOURIER TRANSFORM

## ABSTRACT

Digital watermarking has gained significant attention in various practical applications. It involves embedding specific information into signal data while maintaining the original data quality. Watermarking can be applied to different types of data, including images, video, text, and audio, to protect them from copyright infringement. This thesis presents a few approaches that exhibit unique features and could be suitable for image watermarking which involves embedding a binary image watermark using a combination of the Discrete Wavelet Transform (DWT) and the Fractional Fourier Transform (FrFT) for blind and robust digital watermarking of images. The DWT domain is utilized to ensure the watermark is imperceptible to the human eye. The FrFT orders serve as secret keys that increase the procedure's robustness against various attacks. Additionally, the method enables extraction of the watermark from the watermarked image without the knowledge of the primary image. The objective of this thesis is to examine the efficacy of a technique for watermarking binary images within the wavelet domain. The strategy is evaluated through computer simulations, with particular emphasis on several key issues. These include ensuring the binary graphic watermark is embedded into the original image with minimal distortion and that the watermark can be obtained absent of requirement of the original image. The method is effective and resistant against numerous challenges, including JPEG compression, filtering, noise addition, cropping, and rotation. The proposed method also maintains the confidentiality and integrity of the original image while ensuring that the watermark is imperceptible to the human eye. The study aims to demonstrate the effectiveness and reliability of image watermarking within the wavelet spectrum as a feasible approach for protecting image copyright and ownership, as well as providing authentication and security.

**Keywords:** Watermark, Wavelet Transform, Image Processing, Frequency Domain Transform, Image Watermarking

# KESİRLİ FOURİER DÖNÜŞÜMÜ KULLANARAK SAYISAL İMGE DAMGALAMA

## ÖZ

Dijital damgalama, çeşitli pratik uygulamalarda büyük ilgi görmüştür. Digital damgalama, orijinal veri kalitesini korurken belirli bilgilerin sinyal verilerine gömülmesini içerir. Telif hakkı ihlallerinden korumak için resimler, video, metin ve ses dahil olmak üzere farklı veri türlerine uygulanabilir. Bu tez, benzersiz özellikler sergileyen, kör ve sağlam dijital damgalama için Ayrık Dalgacık Dönüşümü (DWT) ve Kesirli Fourier Dönüşümü'nün (FrFT) bir kombinasyonunu kullanarak resme yerleştirmeyi içeren görüntü damgalaması için uygun olabilecek bir dizi yaklaşım sunmaktadır. DWT alanı, damgalamanın insan görsel sistemi tarafından algılanmamasını sağlamak için kullanılır. FrFT parametreleri, yöntemin çeşitli saldırılara karşı sağlamlığını artıran şifreleme anahtarları olarak işlev görür. Ek olarak, yöntem, damgalama verisinin orijinal görüntü bilgisi olmadan damgalanan görüntüden çıkarılmasını sağlar. Bu araştırmanın amacı, görüntünün dalgacık alanındaki görüntüleri damgalamak için bir tekniğin etkinliğini araştırmaktır. Yaklaşım, birkaç temel konuya özel vurgu yapılarak bilgisayar simülasyonları yoluyla değerlendirilecektir. Bunlar, ikili görüntü filigranının orijinal görüntüye en az bozulmayla gömülmesini ve filigranın orijinal görüntüye ihtiyaç duymadan çıkarılabilmesini içerir. Yöntem, JPEG sıkıştırma, filtreleme, gürültü ekleme, kırpma ve döndürme gibi çeşitli saldırılara karşı etkili ve dirençlidir. Önerilen yöntem aynı zamanda orijinal görüntünün gizliliğini ve bütünlüğünü korurken filigranın insan gözüyle görülmemesini sağlar. Araştırma, görüntü telif hakkını ve mülkiyetini korumanın yanı sıra kimlik doğrulama ve güvenlik sağlama açısından uygun bir yaklaşım olarak dalgacık spektrumu içinde görüntü damgalamanın etkinliğini ve güvenilirliğini göstermeyi amaçlamaktadır.

**Anahtar kelimeler:** Damgalama, Dalgacık Dönüşümü, Görüntü İşleme, Frekans Dönüşümü, Görüntü Damgalama

**CONTENTS**

**Page**

THESIS EXAMINATION RESULT FORM…….. ii
ACKNOWLEDGMENT……..iii
ABSTRACT…….iv
ÖZ…… v
CONTENTS…….vi
LIST OF FIGURES……viii
LIST OF TABLES……... ix

**CHAPTER ONE – INTRODUCTION…… 1**

1.1 Introduction…….. 1
1.2 Digital Watermarking Concepts……...1
1.3 Literature Review……4

**CHAPTER TWO – BACKGROUND……..11**

2.1 Image Watermarking Techniques…… 11
2.1.1 Spatial Domain Watermarking…….. 11
2.1.2 Transform Domain Watermarking…… 12
2.1.2.1 Discrete Cosine Transform (DCT) **……**13
2.1.2.2 Discrete Wavelet Transform (DWT) Based Watermarking**……..** 13
2.1.2.3 Fractional Fourier Transform (FrFT) Based Watermarking**……..**14
2.1.2.4 DWT-FrFT Watermarking**……**15
2.2 Performance Evaluation Metrics……..16
2.3 Wavelet Transform…….. 17
2.3.1 Discrete Wavelet Trsansform……..... 19
2.4 Fractional Fourier Transform (FrFT)……22

**CHAPTER THREE - PROPOSED WATERMARK EMBEDDING AND EXTRACTION METHOD ……...27**

3.1 Watermark Embedding Process…… 27
3.2 Watermark Extracting Process……..32

**CHAPTER FOUR - EXPERIMENTAL RESULTS…….. 36**

4.1 Simulation Results……... 37

**CHAPTER FOUR - CONCLUSION……... 48**

**REFERENCES…….. 50**

**APPENDICES…… 53**

## LIST OF FIGURES

**Page**

Figure 2.1 2D-DWT with 1-level decomposition .....21

Figure 2.2 2D-DWT with 1-level decomposition. .....21

Figure 2.3 2D-DWT with 1-level decomposition. .....22

Figure 2.4 Chirp signal in time domain. .....25

Figure 2.5 Chirp signal in fractional Fourier domain. .....25

Figure 3.1 a) Original image, b) Watermark image .....28

Figure 3.2 Flowchart representation of the method for embedding the watermark .....28

Figure 3.3 a) Original image, b) 2D-DWT with 1-level decomposition .....29

Figure 3.4 a) Lena image, b) magnitude FrFT of the Lena image with order pair (0.88, 1.06), c) inverse transformed image with order pair (-0.56,-1.55), d) inverse transformed image with order pair (-1.46,-0.45).....30

Figure 3.5 Watermarked Lena image .....32

Figure 3.6 Flowchart representation of the method for extracting the watermark .....33

Figure 3.7 a) Original watermark image b) Extracted watermark image.....33

Figure 4.1 Original pepper image. .....36

**LIST OF TABLES**

**Page**

Table 4.1 Extracted watermark images with quality factors, Q, of a) 80%, b) 70%, c) 60%, d) 50%. Gain factor c = 8 is used with chirp signal.....37

Table 4.2 PSNR values of Pepper watermarked images using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values.....38

Table 4.3 PSNR values of Pepper watermarked images using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values.....39

Table 4.4 PSNR values of Lena watermarked images, using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values .....41

Table 4.5 NC values of Lena attacked images watermarked using chirp signals with different block sizes and with a constant gain factor $c = 2$ .....42

Table 4.6 NC values of Pepper attacked images watermarked using PN sequences with different block sizes and with a constant gain factor $c = 10$ .....43

Table 4.7 NC values of Pepper attacked images watermarked using PN sequences with different block sizes and with a constant gain factor $c = 2$ .....44

Table 4.8 C values of Lena attacked images watermarked using PN sequence signals with different block sizes and a constant gain factor $c = 2$ .....45

Table 4.9 NC values of Lena attacked images watermarked using PN sequence signals with different block sizes and with a constant gain factor $c = 2$ .....46

Table 4.10 Extracted watermarks of attacked images with constant gain factor $c = 5$ .....47

# CHAPTER ONE
# INTRODUCTION

## 1.1 Introduction

The widespread use of the Internet has made it easy to copy, modify, and distribute digital images without compromising quality or incurring significant costs. However, this convenience has also brought with it different problems and increased security and privacy concerns for digital data. Protecting data against tampering, copyright infringement, and unauthorized duplication has become a top priority. Digital image watermarking is one of the ways of placing a digital code or image over the original image to protect its ownership, originality and integrity. Depending on the situation or place of use, the watermark can be visible or invisible and serves as a strategy to determine the owner or source of the image. This strategy can be employed in a variety of digital image formats, including photographs, logos, and other graphics. It is often used in digital rights management, copyright protection, and security applications.

After this introductory section, this study is organized as follows. In Section 1.3, we introduce and discuss the related works. Chapter 2 covers the basic image watermarking techniques and discusses the fundamental information regarding the essential prerequisites for digital watermarking methods. Also, we highlight various applications of digital watermarking and provide a concise introduction to different types of watermarking approaches. We emphasize the advantages of implementing watermarking within the wavelet domain. Section 2.2 provides additional explanations and details about the wavelet transform. Section 2.3 contains more detail about Fractional Fourier Transform (FrFt). Finally, in Chapter 3, additional details about our method are introduced, and the results are explained in Section 4.1.

## 1.2 Digital Watermarking Concepts

Digital watermarking is an appraoch to place secret details within digital content, for instance an image, video, or audio clip. Watermarking is used as a way of identifying or tracking content, including copyright protection, authentication, and tamper detection. Watermark can be used in two different ways depending on the

purpose of the work. It can be visible or invisible and can be detected by software tools without damaging the quality of the content.

The history of digital watermarking dates back to the late 1980s when it was first introduced as a method for protecting digital music. Early digital watermarking techniques focused on changing the least significant bits of a digital signal, such as an audio file, to embed a digital code that could be used to identify the content. This technique was vulnerable to attacks, as the watermark could be removed by manipulating the least significant bits without affecting the quality of the content.

In the early 1990s, researchers developed more robust digital watermarking techniques that were resistant to attacks. These techniques were based on the concept of inserting information using the frequency domain of the signal, such as the Fourier Transform (FT), to make it more difficult to remove or tamper with the watermark. Over the years, researchers have continued to develop more advanced and complex digital watermarking techniques using the power of wavelets and neural networks.

There are several types of watermarks used in image watermarking:

Visible Watermarks: These watermarks are intentionally visible and can be used for branding purposes or to indicate ownership of the image. They are usually added as logos or text overlays.

Invisible Watermarks: These watermarks are not visible to the naked eye and are used for copyright protection and authentication purposes. These are inserted into the image using powerful techniques such as spatial or transform domain watermarking.

Fragile Watermarks: A fragile watermark is a type of digital mark that is very sensitive to any changes made to the content it is embedded in. It is called "fragile" because it is easily damaged or removed when the content is modified. The purpose of a fragile watermark is to help verify if digital content has been tampered with or modified without authorization. It is commonly used to protect copyrights, verify the authenticity of data, and assist in digital investigations. Unlike strong watermarks that can withstand various alterations, fragile watermarks are deliberately delicate. They are hidden within the content and are not easily noticeable to the human eye. This

makes them hard to remove without leaving visible traces or affecting the content in some way. When the watermarked content is changed, even small modifications like resizing or color adjustments, the fragile watermark gets distorted or destroyed. By comparing the modified content with the original, the presence of alterations can be detected, helping to ensure the integrity of the content. Fragile watermarks are widely used in digital image forensics to determine if an image has been manipulated. They can be used as important tools to confirm the authenticity of digital media along with maintaining the integrity of important documents and sensitive information.

Robust Watermarks: A robust watermark is a digital mark that remains detectable even when the content it is applied to undergoes changes. Unlike fragile watermarks, robust watermarks can withstand modifications without losing visibility. These watermarks are designed to withstand common image processing techniques and attacks. They are used to protect images from unauthorized use and distribution and used for copyright protection, content authentication, and digital rights management. Robust watermarks resist alterations and can survive image manipulations and compression techniques. Through measuring the procured watermark against the initial, the originality of the content and document integrity can be verified. They are widely used in digital media to protect intellectual property and ensure ownership of digital assets.

Blind, semi-blind, and non-blind watermarks are classified dependent on the watermark recovery method. In blind watermarking, you can uncover the watermark without needing the original, unaltered image or video. Only a secret key is necessary for this process. Blind watermarking is regarded as the most robust type of watermarking due to its independence from the original content. This feature makes it highly resistant to tampering. While it offers strong security, it typically involves complex mathematical techniques, making it more challenging to implement. Semi-blind watermarking is like finding a middle path between security and practicality. To get the watermark back, you need the secret key and the modified image or video. It provides a level of protection as the private key is required for watermark extraction. Semi-blind watermarking is more user-friendly than blind techniques, making it a practical choice for various applications. Non-blind watermarking, often referred to as

traditional watermarking, relies on accessing the original, unaltered image or video to recover the content. It is placed directly inside the original file. It is the least secure approach, as an unauthorized user with access to the original content can easily tamper with or remove the watermark. Non-blind watermarking is the simplest to implement, employing straightforward embedding and extraction methods. This is how basic copyright protection can be achieved. In addition, it is widely used in content authentication.

The choice of watermarking technique depends on specific application requirements and trade-offs between security, complexity, and ease of implementation.

## 1.3 Literature Review

Image watermark techniques are generally divided into two, one of which is the spatial domain technique, whereas the second one is the transformation domain technique. Embedding watermark by modifying the pixels in an image is a spatial domain technique (N. Nikolaidis, & I. Pitas, 1998). Alternatively, wtihin the scope of transform domain watermarking, the procedure starts by transforming into the underlying transform domain. Discrete Cosine Transform, Fourier Transform (FT), Discrete Wavelet Transform (DWT) are different types of transforms. Recently, the fractional Fourier transform (FrFT), as an alternative frequency domain technique, has also been used for watermarking (I. Djurovic, S. Stankovic, & I. Pitas, 2001). The study (I. Djurovic, S. Stankovic & I. Pitas, 2001) uses white Gaussian noise as a watermark where white Gaussian noise has been applied for both embedding and detection. The transform orders of the FrFT and pixel positions where watermark is embedded are utilized together as encryption keys. F. Q. Yu, Z. K. Zang and M. H. Xu used a linear chirp signal, which creates a sharp peak in the FrFT domain if the correct transform order is used, is employed as the watermark and the data is inserted in the FrFT domain.

I. Cox, J. Killian, F. T. Leighton, and T. Shamoon (I. Cox, J. Killian, F. T. Leighton, & T. Shamoon,1997), proposed a method involving watermarking performed in spatial domain which is robust to attacks. The authors use a two-step process where the

watermark is first placed on the image by a secret key and then verified using the essential data key to detect and decode the watermark. The method uses the power of the spread spectrum approach, where the watermark is spread across the image using a pseudorandom sequence. The authors utilize the least significant bit (LSB) of the data to embed the watermark, ensuring that the watermark is imperceptible to the human eye. It works by first splitting the image into several areas. The watermark is then placed in each block using a spread spectrum technique. The watermark spreads throughout the block so that it is not easy to detect or eliminate without compromising the visual integrity. The watermark is also expected to be resistant to noise and other distortions. The method was evaluated by embedding watermarks into a variety of images and then subjecting the images to a variety of attacks. The results showed that it can be used to embed watermarks that are resistant to a variety of attacks including image compression, cropping, and resampling. The method was also robust to noise and other distortions. The authors have successfully validated the efficiency of their suggested method through its application on digital image audio signals and video sequences. They have also evaluated the method's resilience to different types of disruptions, including data compression, image filtering, and alterations in shape and perspective. The findings from the experiments indicate that the method is exceptionally resilient and secure. It demonstrates an ability to resist numerous types of disruptions while preserving both the integrity and the undetectability of digital images.

The method proposed by Solachidis and Pitas (V. Solachidis, & I. Pitas, 2001), relies on the modification of the phase spectrum of the 2D-DFT of the original image, while the watermark extraction relies on the correlation between the phase spectrum of the watermarked and the original images. The circularly symmetric watermark is inserted into the 2D discrete Fourier transform (DFT) domain of an image. The authors use a circular mask to limit the watermark embedding to the central region of the 2D-DFT domain. This approach is based on the observation that most of the the image's energy resides within the low-frequency components of the 2D-DFT. By embedding the watermark in this region, the watermark becomes less susceptible to image processing operations that affect high-frequency coefficients. The proposed watermark embedding algorithm first generates a watermark pattern in the spatial domain, which

is then transformed into the frequency domain using a 2D-DFT. The watermark pattern is scaled and thresholded using a circular mask that limits the watermark embedding to the central region of the 2D-DFT domain. The resulting watermark spectrum is then added to the host image's 2D-DFT spectrum in the same circular region. Ultimately, watermarked images are produced via carrying out inverse two-dimensional DFT. Watermark detection algorithm that works by first estimating the circular mask's parameters is used during the watermark embedding process. Subsequently, the watermark's existence is confirmed via comparing thresholded circularly symmetric correlation of watermarked image's 2D-DFT and the watermark pattern's 2D-DFT. The watermark is placed into the 2D-DFT elements of the image in such a way that the watermarked image is still circularly symmetric. This means that the image is still rotationally invariant, namely, it still looks the same when it is turned through any degree of rotation. This property makes the watermarked image more resistant to attacks such as resampling and trimming. These attacks typically involve rotating the image, and if the watermarked image is still circularly symmetric after the attack, then it is more likely that the watermark will still be intact.

The paper by D. Kundur and D. Hatzinakos (D. Kundur, & D. Hatzinakos, 1997), proposes a robust digital image watermarking method based on wavelet-based fusion. The method is designed to insert the watermark within the low frequency sub-bands of image, enhancing robustness against typical operations including compression, filtering, and noise addition. The suggested approach operates in the following manner: The initial host image is broken down into its wavelet sub-bands through a wavelet transformation. The low frequency sub-bands are then selected for watermark embedding because they are more perceptually important and more resistant to noise and other distortions. The watermark is placed into these sub-bands through a merging method. This method blends the watermark signal with the original signal to optimize the signal-to-noise ratio (SNR) of the watermark.

Another image watermarking method by Dugad R., Ratakonda K., and Ahuja N. (R. Dugad, K. Ratakonda, & N. Ahuja, 1998), also utilizes wavelet transform. The method employs the discrete wavelet transform (DWT) to break down the source image into multiple frequency bands and watermark is placed inside the bands with

lower frequencies. Authors employ a sequence that is both pseudo-random and dependent on a key to adjust the watermark prior to its integration into the source image. The watermark is placed to chosen sub-bands using a simple addition operation. The authors also suggest a detection method to retrieve the watermark from the image that has been watermarked. This detection method initially employs the DWT on the watermarked image to acquire the bands and initial watermark sequence to identify the presence of watermark.

The paper by I. Djurovic, S. Stankovic and I. Pitas (Djurovic I., S. Stankovic, & I. Pitas, 2001), proposes a new method for digital watermarking that uses the fractional Fourier transform (FrFT). In this paper, they examine the concept of image watermarking within the realm of the FrFT. This method utilizes a blend of the spatial and spatial-frequency domains. They suggest a digital watermarking approach for images that relies on FRFT which is employed to modify image for two angles, ax and ay, and the transformation coefficients that result are rearranged in a sequence that does not increase. The smallest coefficients are disregarded, and the watermark is inserted into the subsequent group of transformation coefficients. This strategy prevents considerable image distortion that could happen if watermark is inserted into the larger data, and it prevents the watermark from being easily removed by lossy image compression or lowpass filtering if placed inside the smaller coefficients. To retrieve watermarks, watermarked image is subjected to the inverse FrFT, and correlation is employed to identify the watermark. Detection of watermark depends on whether watermark key and the locations of the transformation coefficients are known. Gaussian white noise with certain variance is selected as the watermark. The value obtained from watermark detection is compared against a certain limit to confirm the existence of the watermark. The variance of watermark key is determined by balancing invisibility of watermark and the likelihood of incorrect detection.

The paper by Yu, Zang and Xu (Yu F. Q., Z. K. Zang, & M. H. Xu, 2006), introduces a novel approach for watermarking images by using a chirp signal as the watermark and incorporating it into FrFT domain of the image. This method merges data encryption and digital watermarking to boost the safety of the watermark. By incorporating the watermark into certain coefficients in the FrFT domain, the

technique guarantees that it is invisible to the human eye while also offering a high degree of protection. The sequence of the transforms and location of data are utilized as secret keys which further enhances the security of watermark. To detect watermarks, researchers took advantage of the unique property of chirp signals which can produce a strong impulse when the sweep rate matches the FrFT. This approach eliminates the need for selecting a threshold which can lead to errors in watermark detection. The outcomes of numerous assault experiments show that the suggested algorithm is extremely efficient and resilient to JPEG compression, noise intrusions, cropping, and filtering.

The paper by Zhang Feng, Mu Xiaomin and Yang Shouyi (Zhang F., Mu X., & Yang S., 2005), proposed a watermark embedding technique involving embedding chirp signals directly into the original image. This specific chirp signal, when used as a watermark, concentrates energy at a particular order in the FrFT domain, unlike the original image which does not exhibit such energy concentration. Consequently, even if the watermark is weak, its energy stands out above the original image's energy density at the specific transform order. This property is useful for detecting the chirp watermark. By identifying the maximum value of the squared modulus in the vicinity of the specific transform order, the presence of the watermark can be detected.

The paper by Wang Dian-hong, Li Dong-ming, Yan Jun, and Chen Fen-xiong (Wang D., Li D., Yan J., & F. Chen, 2007), outlines a robust digital watermarking method for enhancing copyright protection, verifying sources, and ensuring data integrity. This method involves embedding a two-dimensional chirp signal watermark in the host image's low-frequency band using wavelet transform. Parameters of the chirp signal are used as keys for extracting the watermark. The algorithm employs the FrFT on the low-frequency band, increasing the frequency step by step to scan the image. By identifying local peaks in the resulting graph, the embedded chirp signal watermark can be detected without requiring the original image for extraction.

Lei Gao, Lin Qi, Yongjin Wang, Enqing Chen, Shouyi Yang and Ling Guan (Lei Gao, Lin Qi, Yongjin Wang, Enqing Chen, Shouyi Yang, & Ling Guan, 2012) present an innovative method for digital image watermarking that combines a two-dimensional (2D) chirp signal with the invariant properties of addition and rotation

with respect to the 2D-fractional Fourier transform (2D-FrFT) to boost the robustness and security of watermark in the 2D-FrFT domain. This method utilizes the varying frequency characteristic of the 2D chirp signal in the watermark, making it resistant to stationary filtering — filters that have constant parameters throughout the image — and ensuring geometric symmetry. As a result, the relative position of the peak in the FrFT domain remains unchanged even after translation and rotation. A noticeable peak in the detection results indicates the presence of the watermark signal. Given that there are two transform orders used, two peak values are expected to be observed in the detection process.

Mohammed Baziyada, Ibrahim Kamelb, Tamer Rabieb and Grigory Kabatyansky (Mohammed B., Ibrahim K., Tamer R., & Grigory K. Xu, 2023),present a new method for protecting Deep Neural Network (DNN) models from unauthorized copying by embedding digital watermarks. The key idea is to achieve robustness by embedding a large watermark that covers the entire model. The process starts by extracting the DNN model's weights and arranging them into a one-dimensional vector. Then, the Discrete Cosine Transform (DCT) is applied to this vector. Embedding a large watermark can negatively impact the DNN model's performance. To balance performance, capacity, and robustness, the method uses the DCT because of its strong ability to compact data, especially when applied to related segments of data. Thus, the technique involves extracting and grouping the DNN weights into correlated segments before applying the DCT. Randomizing the data before embedding helps spread the watermark signal throughout the DNN model, ensuring that if one part of the model is compromised, the watermark can still be retrieved with sufficient quality. The watermark is embedded by altering the insignificant DCT coefficients. After embedding, the inverse DCT is applied, and the adjusted weights are placed back into their original locations within the DNN model.

Yimeng Zhao, Chengyou Wang, Xiao Zhou and Zhiliang Qin (Yimeng Zhao, Chengyou Wang, Xiao Zhou, & Zhiliang Qin, 2023), introduces a method for robust image watermarking using deep learning and attention network. The framework consists of four key components: an attention network, a watermark embedding network, a watermark extraction network, and an attack layer. The attention network,

known as the channel, and spatial attention network calculate attention weights in two dimensions assigning different weights to pixels based on their channels and positions. This is applied during both the watermark embedding and extraction phases.

The attention network, through end-to-end training, can identify areas of the image that are less noticeable to the human eye and apply higher weights to these areas during the watermark embedding process. This enhances the watermark's invisibility. In the watermark extraction phase, the loss function is designed to give greater emphasis to features containing the watermark while minimizing the impact of noise, enabling the extraction network to focus on watermark-related features and reduce noise from attacks, thereby enhancing robustness.

To address issues like gradient vanishing or explosion in deep networks, residual modules are incorporated into both the embedding and extraction networks. Experiments indicate that this approach can embed watermarks without affecting how people perceive the image, while also providing strong resistance against various attacks.

# CHAPTER TWO
# BACKGROUND

## 2.1 Image Watermarking Techniques

Image watermarking methods consist of incorporating a digital watermark into image to safeguard the ownership, genuineness, or integrity of the image. These techniques can be seperated into two main groups: Spatial domain techniques and transform domain techniques. In spatial domain techniques, the watermark is embedded straight into the image's pixels, usually by subtly altering color or brightness levels of chosen pixels. Transform domain techniques, on the other hand, involve converting initial image into a seperate domain such as the Fourier transform (FT) domain, the Discrete Cosine Transform (DCT) domain, the Discrete Wavelet Transform (DWT) domain, and afterwards integrating watermark into the transformed coefficients.

These methods can also be separated as robust and fragile watermarking techniques. Robust watermarking methods are utilized to incorporate watermarks that can endure typical image processing tasks, such as resizing or compression, while fragile watermarking methods are employed to incorporate watermarks that can identify even slight modifications to the image.

### 2.1.1 Spatial Domain Watermarking

Methods that entail directly incorporating a watermark onto pixel values of the image are classified as spatial domain watermarking. That is to say, the watermark is integrated inside the spatial domain of the image. In this method, watermark is directly placed onto the image by altering color or brightness values of specific pixels or pixel groups. Spatial domain watermarking can be further divided into two subcategories: Insertion of the least significant bit (LSB) and spread spectrum watermarking.

LSB insertion is a process where each chosen pixel is substituted within corresponding bit from watermark data. In the modification of the chosen pixel, the LSB of randomly chosen pixels is changed to embed a watermark by replacing it with the most significant bit (MSB) of a different signal. This procedure generates a random

signal using a specific key. Watermark is then integrated into the LSBs of the original image and can be retrieved in a similar manner.

The alternative method is called spread spectrum watermarking. Instead of embedding the watermark in the least significant bits of the data, this method utilizes principles from spread spectrum communication. The watermark is dispersed over numerous frequency bins, ensuring that the energy in any single bin is minimal and unnoticeable. This increases the difficulty for an attacker to eliminate or adjust the watermark without maintaining the image's integrity. Additionally, the watermark is designed to resemble additive random noise, which makes it even more difficult to detect.

The spread spectrum watermarking technique proposed by Cox, Killian, Leighton and Shamoon, in 1997, is effective against a range of attacks, including cropping, comppression, and resizing. It is also resistant to collusion attacks, where multiple attackers collude to remove or modify the watermark.

### 2.1.2 Transform Domain Watermarking

Transform domain techniques modify the coefficients of an image or signal in transform domain, including the Fourier Transform (FT) or the Discrete Wavelet Transform (DWT). The technique involves transforming the original data into a different domain, modifying certain coefficients to embed the watermark, and then transforming the modified data back to the original domain (N. N. Nweke, A. M. Alhassan and T. M. Agu, 2013). In transform domain watermarking, watermark is inserted into specific elements of the signal, which can improve robustness and unnoticeability of watermark compared to spatial domain techniques. However, watermarking in the transform domain may also introduce some perceptual distortions or artifacts in the original signal.

Overall, transform domain watermarking is a powerful technique for copyright protection and verifying content authenticity, particularly when combined with other techniques such as encryption and digital signatures.

#### *2.1.2.1 Discrete Cosine Transform (DCT)*

DCT based watermarking is a digital watermarking approach that embeds watermark into a source image or video by altering DCT coefficients of the image or video. DCT is a numerical method that transposes signals or visual representations from spatial domain to spectral domain, allowing for efficient compression and processing.

The fundamental concept underlying DCT-based watermarking is to place watermarks by modifying the DCT coefficients of the host signal. Modification of the DCT coefficients is done by adding or subtracting a small amount of data to the selected DCT coefficients, which are chosen based on their sensitivity to small modifications (M. Barni, F. Bartolini, V. Capellini & A. Piva, 1998). The watermark can be inserted into the DC coefficients which represent the average brightness of the block, or into the AC coefficients which represent high-frequency elements of blocks.

The DCT-based watermarking process consists of the following stages: Divide host signal into blocks of equal size; apply the DCT to each block; modify DCT coefficients to incorporate the watermark; and then utilize the inverse DCT to acquire the signal that has been watermarked. One of the advantages of DCT-based watermarking is that it is resistant to typical operations in signal processing such as compression, filtering, and cropping. This is because the watermark is placed in the frequency domain, which is less affected by these operations than the spatial domain. However, DCT-based watermarking is susceptible to attacks that specifically target the DCT coefficients, such as geometric attacks or noise attacks.

#### *2.1.2.2 Discrete Wavelet Transform (DWT) Based Watermarking*

Watermarking based on DWT is a kind of watermarking method in the transform domain that employs DWT to incorporate a watermark into an image. DWT breaks down an image into different frequency and resolution sub-bands, enabling the watermark to be placed into the high-frequency sub-bands that are less noticeable to human eye.

In watermarking that utilizes DWT, watermark signal is initially transposed using DWT, following which the derived coefficients are incorporated into chosen sub-bands of the primary image. Over the course of the detection phase, the DWT is employed on the image with the watermark to retrieve the watermark signal, which is subsequently contrasted with the original watermark to ascertain if there have been any alterations.

The first step in DWT-based watermarking is partitioning the image into disjoint segments of equal size. Then, DWT is applied to each segment of the image. This transposes the image into various frequency and resolution subgroups. The watermark is then embedded into the high-frequency sub-bands of the DWT coefficients. This is done via altering the DWT coefficients of high-frequency sub-bands to embed watermark information. After the watermark has been incorporated, the DWT elements are employed to reassemble image. To retrieve the watermark, DWT is utilized on images that has been watermarked to derive the DWT coefficients. The watermark is subsequently extracted from the altered high-frequency subgroups.

Watermarking that utilizes DWT has gained widespread adoption due to its capacity to offer high unnoticeability and resilience against a variety of image processing operations. It finds use in a diversity of applications, including safeguarding copyright, verifying authenticity, detecting tampering, and concealing data. It is deemed to be robust in the face of various forms of assaults, including the addition of noise, filtering, and compression.

#### *2.1.2.3 Fractional Fourier Transform (FrFT) Based Watermarking*

FrFT-based watermarking is a technique that uses FrFT to place watermark signal into a digital image. As a generalization of Fourier transform (FT), the FrFT can be employed for signal analysis in a different domain than the time or frequency domains (L. B. Almedia, 1994).

In FrFT-based watermarking, the watermark signal is initialy transformed into the FrFT domain, and after that the transformed watermark signal is placed into the image by modifying the FrFT coefficients. The embedding process can be achieved by adding or subtracting a fraction of the signal from the original image elements.

Same FrFT operation is applied to the watermarked image to retrieve the watermark image and the watermark signal can be obtained by subtracting FrFT of the initial image from FrFT of the image that has been watermarked.

FrFT-based watermarking has been studied in various research works, and it has shown promising results with respect to robustness and security. However, it also has some limitations, such as the sensitivity to the choice of the FrFT orders and the computational complexity of the FrFT operation.

#### *2.1.2.4 DWT-FrFT Watermarking*

DWT-FrFT watermarking is a technique that combines the advantages of DWT-based and FrFT-based watermarking. In this method, the initial image undergoes a transformation using DWT, followed by a further transformation of low-frequency sub-band using FrFT. Watermark is incorporated into the transformed coefficients utilizing a confidential key. This method yields a more resilient and visually unnoticeable watermark, thereby complicating the task for attackers to eliminate or modify the watermark (O. Duman and O. Akay, 2011).

The subsequent procedures are involved in the hybrid DWT-FrFT watermarking technique: i) Take an initial image and implement DWT to obtain high-frequency sub-bands. ii) Apply FrFT to the high-frequency sub-bands. iii) Embed the watermark in the FrFT domain utilizing confidential keys. The location of the embedded watermark and orders of transformation make up the confidential key. iv) Combine the watermarked high-frequency sub-bands with the low-frequency sub-bands obtained in the first step.

Inverse DWT is applied to retrieve watermarked images to identify the watermark. The watermarked image undergoes a transformation via FrFT utilizing the identical confidential key that was applied during the embedding phase. Subsequently, the watermark is retrieved from the FrFT domain. Finally, the extracted watermark is decrypted using the same confidential key applied during the embedding phase. If the original watermark is identical to the watermark that has been decrypted, the image is considered authentic. These steps can be repeated for embedding multiple watermarks with different keys in different sub-bands.

## 2.2 Performance Evaluation Metrics

Performance evaluation metrics are used to measure the effectiveness of image watermarking techniques. Some of the common techniues are Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), Normalized Correlation (NC), Structural Similarity Index (SSIM), and Bit Error Rate (BER).

The formula below is applied to calculate value (in dB) of the PSNR which evaluates excellence of a watermarked image with 256 grayscale intensity values.

$$PSNR = 20\,log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \tag{2.1}$$

In the pixel domain, MSE quantifies discrepancy between watermarked image and the original image. MSE between the NxN original image, p(x, y), and the watermarked image, p(x,y)', is defined as

$$MSE = \frac{1}{N\times N}\sum_{x=1}^{N}\sum_{y=1}^{N}[p(x,y) - p(x,y)']^2 \tag{2.2}$$

The measure of similarity between initial and extracted watermarks, based on the correlation value, is evaluated by Normalized Correlation (NC). It is defined as follows,

$$\rho = \frac{\sum_{x,y=1}^{N}(w(x,y)-\bar{w})(w^w(x,y)-\bar{w^w})}{\sqrt{(\sum_{x,y=1}^{N}(w(x,y)-\bar{w})^2)(\sum_{x,y=1}^{N}(w^w(x,y)-\bar{w^w})^2)}} \tag{2.3}$$

SSIM evaluates likeness between two signals considering aspects such as brightness, contrast, and structural variation.

BER quantifies rate of error of watermark that has been extracted in relation to the original watermark.

Robustness measures the ability of the watermarking technique to resist various image processing attacks such as cropping, filtering, and compression.

Capacity measures the optimum data capacity that can be embedded in the image without affecting its perceptual quality.

Computational complexity measures the processing time and complexity required to embed or retrieve the watermark.

The above metrics can be utilized to compare and evaluate the effectiveness of various image watermarking techniques.

## 2.3 Wavelet Transform

Wavelet transform is a mathematical method utilized in signal processing, data compression, and numerous other disciplines. It represents a relatively recent advancement when compared to traditional signal processing approaches like FT. A significant landmark in the history of wavelet transforms was the development of Mallat algorithm, or discrete wavelet transform (DWT), by Stephane Mallat, 1989. This algorithm brought a substantial boost in the efficiency of wavelet-based signal processing.

Wavelet transforms have gained popularity across a broad spectrum of areas including image compression and noise reduction. They have also found their way into fields such as signal processing, data analysis, and image processing. The publication, ‘‘A Wavelet Tour of Signal Processing’’, by Stephane Mallat played a pivotal role in popularizing wavelet theory for those interested in the field.

Over time, wavelet transforms have continued to evolve, leading to the development of various wavelet types including biorthogonal and complex wavelets. Their applications have expanded into areas such as medical imaging, data compression, and feature extraction in machine learning.

Wavelet transforms are now a fundamental component of signal processing and data analysis. They offer distinct advantages over traditional Fourier analysis, especially when dealing with signals exhibiting non-stationary characteristics, as they supply both time and frequency information in a localized manner. Consequently, wavelet transforms have been applied across a wide spectrum of disciplines, from image and audio processing to financial data analysis and beyond. Their adaptability and versatility make them an invaluable tool in modern data analysis and signal processing.

A wavelet is a fluctuation that resembles a wave and is confined in time. Wavelets possess two fundamental characteristics: Scale and location. The scale delineates the extent of the wavelet's stretch in relation to the frequency, as specified for waves. The location determines the temporal (or space) positioning of the wavelet. When the value of scale is decreased, the wavelet becomes more compressed, allowing it to capture higher frequency information. On the other hand, when scale is increased, the wavelet becomes more stretched, making it better suited for capturing lower frequency information.

If the position or location of the wavelet decreases, the wavelet will shift towards the left, while increasing its value will shift the wavelet towards the right. Location is a essential characteristic of wavelets, given that they only exhibit non-zero values within a limited interval, contrasting with waves. Furthermore, when analyzing a signal, it's not just vital to observe its fluctuations, but also the specific points in time or space where these fluctuations take place. Therefore, the location property of wavelets is crucial for accurately representing signals in time and frequency domains.

Wavelet Transforms primarily come in two varieties: the Discrete Wavelet Transform, also known as DWT, and the Continuous Wavelet Transform, referred to as CWT. CWT utilizes an infinite number of wavelets at different scales and locations to analyze a signal, while the DWT uses a limited collection of wavelets at specific scales and locations to break down a signal into sub-bands. In other words, the CWT analyzes a signal at every possible scale and location, while the DWT only analyzes a signal at a predetermined set of scales and locations.

The wavelet transform of a continuous-time signal x(t) can be expressed as follows

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\, \psi^*(\frac{t-b}{a}) dt \quad (2.4)$$

where $\psi^*(t)$ represents the complex conjugate of the wavelet function ψ(t), a represents dilation parameter, and b denotes location parameter.

### *2.3.1 Discrete Wavelet Trsansform*

DWT is a transform which is used to analyze signals and extract their frequency components. It is a multi-resolution analysis technique that decomposes a signal into a series of wavelet coefficients at varying scales and positions. In mathematical terms, DWT represents a constant, unchanging linear transformation that is applied to a discrete-time signal or sequence.

The formula for the DWT can be written as

$$T_{m,n} = \int_{-\infty}^{+\infty} x(t)\, \psi_{m,n}(t) dt \quad (2.5)$$

where $T_{m,n}$ is referred to as the wavelet (or detail) coefficient at the indices of scale, $m$, and location, $n$.

The transform is based on a set of wavelets which are small functions that are scaled and translated to cover the entire signal. DWT is predicated on the breakdown of signal into collection of subsidiary signals, or sub-bands, via a collection of low-pass and high-pass filters. The low-pass filter singles out the low-frequency components of the signal, whereas the high-pass filter singles out the high-frequency components. This procedure is then iteratively applied to the low-pass sub-band, resulting in a tree-like structure of sub-bands.

DWT decomposes the input signal into a set of approximation coefficients, C(j,k), and detail coefficients, D(j,k), at different scales and positions, which can be used for various signal and image processing applications including compression, denoising, and feature extraction.

DWT is commanly used in many applications such as image processing, audio processing, data compression, data analysis, denoising, feature extraction, and signal analysis.

In image processing, DWT is applied to a two-dimensional signal represented by the image to acquire a set of coefficients in the form of sub-bands. These sub-bands are essentially a set of images, each capturing a different level of detail and resolution of the original image. The decomposition can be performed in two dimensions, resulting in a set of four sub-bands at each level; one approximation sub-band and three detail sub-bands. The approximation sub-band represents low-frequency content of the image, while detail sub-bands capture high-frequency details in different directions (see Figures 2.1, 2.2, and 2.3). Note that the LL sub-band represents the low-frequency information (approximation) of the image, while the HL, LH, and HH sub-bands represent high-frequency information (details) of the image in different directions.

The DWT Haar transform can be computed iteratively to obtain a multi-level decomposition of the signal. The decomposition level is determined by the number of times the decomposition step is repeated. The reconstructed signal should be equal to the original input signal.

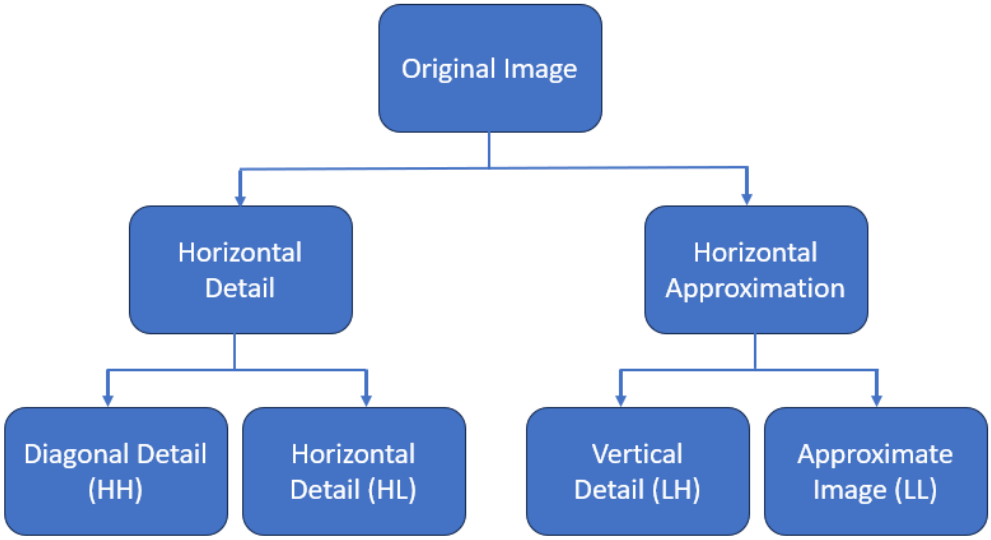



Figure 2.1 2D-DWT with 1-level decomposition

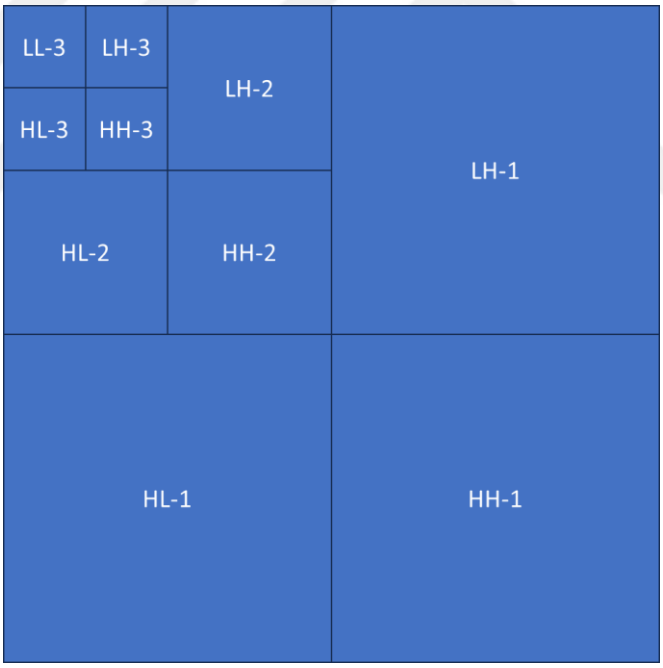



Figure 2.2 2D-DWT with 1-level decomposition



Figure 2.3 2D-DWT with 1-level decomposition

## 2.4 Fractional Fourier Transform (FrFT)

FrFT is a transform that generalizes classical Fourier transform by introducing a parameter that allows for the selection of a specific order or degree of the transform. The FrFT is defined as a linear transformation that converts a time-domain signal into a frequency-domain representation. In contrast to traditional Fourier transform, which yields a static breakdown of a signal into its frequency constituents, the FrFT permits a flexible dissection of the signal into its frequency elements.

The FrFT is a potentially applicable tool for processing especially non-stationary signals such as linear chirps. The FrFT domain can be considered as a sort of fusion of the time and frequency domains. Within the time-frequency dimension, the classical FT is asociated with a rotation angle $\frac{\pi}{2}$ which is a special case of all the FrFT domains [8,9]. As characterized by the transformation angle, $\alpha$, the one-dimensional FrFT can be defined as

$$\mathrm{IF}^{\alpha}\{f(t)\}(u) = \int_{-\infty}^{\infty} f(t)K^{\alpha}(t,u)dt \qquad (2.6)$$

with the transformation kernel given as

$$K^{\alpha}(t,u) = \sqrt{\frac{1-j\,cot\,\alpha}{2\pi}}\, e^{j(t^2+u^2)\frac{cot\,\alpha}{2} - jtu\,csc\,\alpha} \tag{2.7}$$

The inverse of the FRFT can be obtained by taking an FRFT with the negative transformation angle, -α. Thus, we can write

$$f(t) = \mathrm{IF}^{-\alpha}\{\mathrm{IF}^{\alpha}[f(t)]\}(t) \tag{2.8}$$

Two-dimensional FrFT is used for image analysis with a transform order[1] pair $(a_1, a_2)$ which are implemented on the rows and columns of a picture [30]. It is a straightforward generalization of the one-dimensional FrFT in (2.6) for two-dimensional signals, namely images.

Computation of 2D-FrFT of the image, along with the inverse transform, can be expressed as follows

$$F_{\alpha,\beta}(m,n) = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} f(p,q)\, K_{\alpha,\beta}(p,q,m,n) \tag{2.9}$$

$$f(p,q) = \sum_{p=0}^{M-1}\sum_{q=0}^{N-1} F_{\alpha,\beta}(m,n) K_{-\alpha,-\beta}(p,q,m,n) \tag{2.10}$$

A significant characteristic of the FrFT is its capacity to alter the orientation of the transformation, which means that it can be used to transform signals in the time-domain, frequency-domain, and other intermediate domains. The FrFT is a parametric

[1] The FrFT order, $a$, and the corresponding transform angle, $\alpha$, are related as $a = (2\alpha/\pi)$.

transform, and the choice of the transform parameter α determines the degree of the transform and the resulting frequency decomposition.

FrFT has several useful properties, including the ability to perform filtering, compression, and data hiding operations. In particular, FrFT can be used for watermarking applications, where a watermark signal is placed into an original signal by altering the transform coefficients. The FrFT-based watermarking schemes have numerous benefits compared to other transform-based watermarking methods, including increased robustness towards signal manipulation assaults on the picture such as compression and filtering.

FrFT has chirp signals as its basis functions. A chirp signal is a type of signal whose frequency either increases or decreases over time. It is also referred to as a signal with linear frequency modulation (LFM). The equation for a continuous-time chirp signal can be written as

$$s(t) = e^{j2\pi(f_0 t + \frac{1}{2}mt^2)} \tag{2.11}$$

where $s(t)$ is the signal at time $t$, $f_0$ is the starting frequency, and $m$ is the chirp rate (see Figure 2.4). A chirp signal can be precisely located if the order of the FrFT is appropriately selected. The base function of the transformation which has the same sweep rate as the chirp signal will yield a narrow peak in the amplitude spectrum of the transformation (see Figure 2.5).

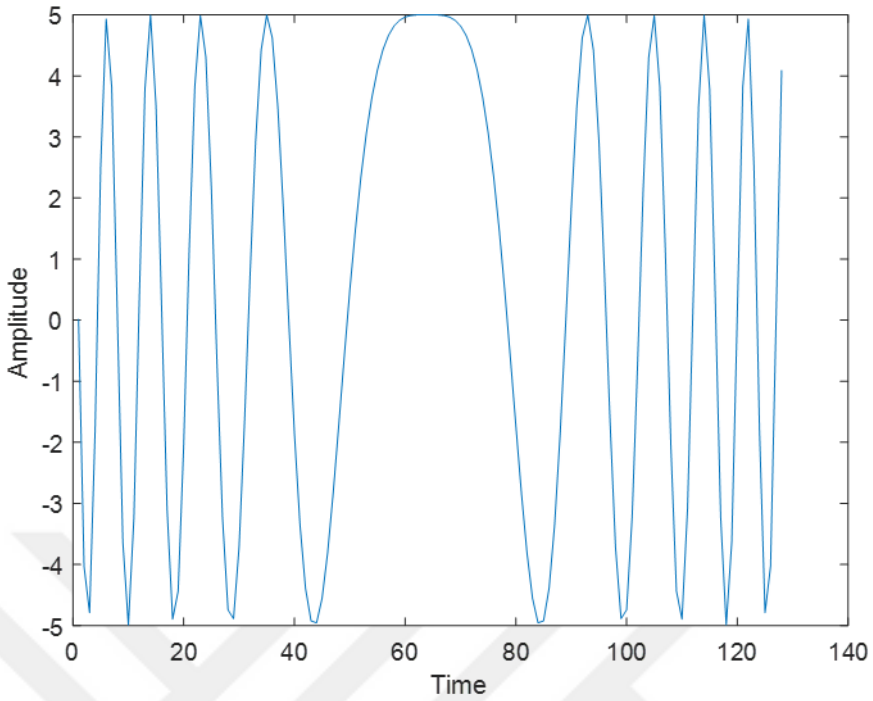



Figure 2.4 Chirp signal in time domain

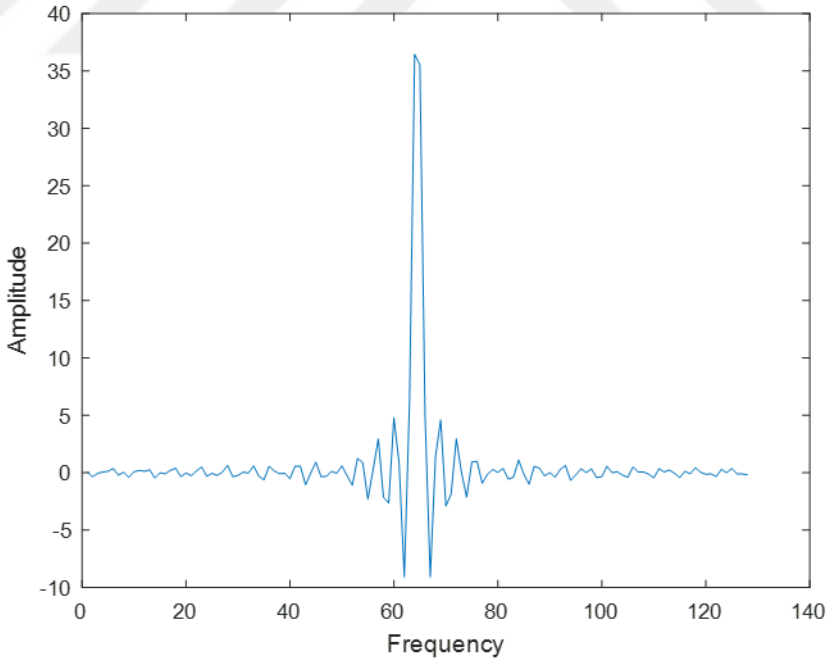



Figure 2.5 Chirp signal in fractional Fourier domain

Chirp signals have many applications in signal processing, including radar, sonar, and communication systems. They are also used in medical imaging, such as ultrasound, to create images of internal organs and tissues.

# CHAPTER THREE
# PROPOSED WATERMARK EMBEDDING AND EXTRACTION METHOD

This study describes the method for blind processes of embedding and detecting digital watermarks in images. The logo, which is a binary image, is integrated into the original image using a blend of DWT and FrFT. DWT is employed to incorporate the watermark in a manner that remains undetectable to the human eye. FrFT orders serve as encryption keys, enhancing the resilience of the watermarking technique against assaults. The extraction phase can proceed without the requirement for knowledge of original image, and the method can withstand various attacks.

The proposed method involves two steps: Embedding the watermark and extracting the watermark. In the embedding phase, the watermark is placed by modifying the elemenst in the transform domain, specifically in the DWT domain. This ensures that the watermark is imperceptible to the human eye. The FrFT is utilized to cipher the watermark, thereby increasing the complexity for intruders to eliminate or meddle with the watermark.

During the extraction phase, watermarks can be identified and retrieved from processed image, even in the absence of original image. This implies that the watermark can serve multiple functions such as safeguarding copyright, verifying authenticity, or detecting tampering. The suggested technique is engineered to withstand a variety of attacks, thereby enhancing its ability to preserve the watermark's integrity.

## 3.1 Watermark Embedding Process

The schematic illustration of our suggested approach for watermark insertion along with the original image and the watermak can be seen in Figures 3.1 and 3.2. To carry out the embedding process, we first break down the initial image intended for watermarking (in our case, a 512x512 image) into four distinct wavelet sub-bands, each characterized by different frequency ranges.



(a) (b)

Figure 3.1 a) Original image, b) Watermark image

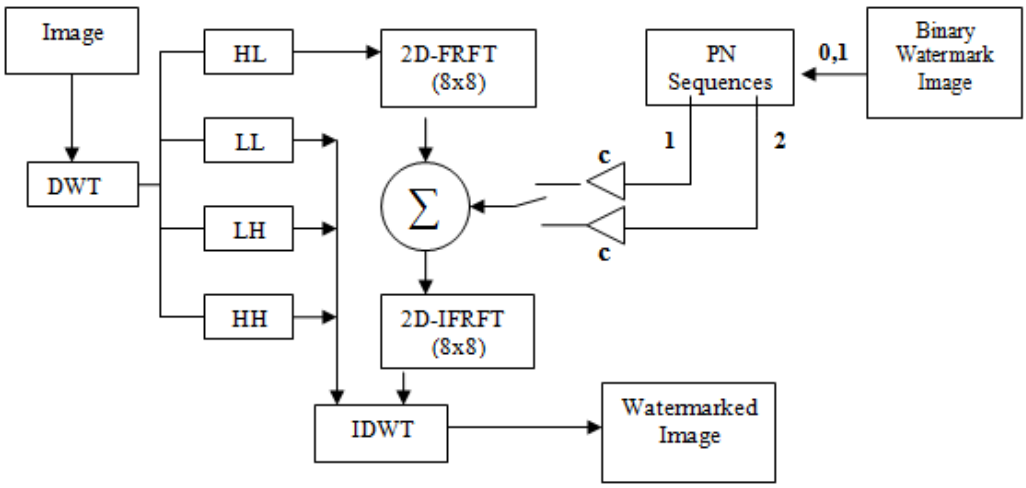



Figure 3.2 Flowchart representation of the method for embedding the watermark

DWT is a technique that breaks down an image into sub-bands, each containing information such as approximation, vertical, horizontal, and detail components. Embedding the watermark in the high-frequency sub-band offers improved invisibility, on the other hand, the low-frequency sub-band exhibits greater resilience to image processing assaults such as low-pass filtering and “jpeg” compression.

The high-low sub-band is chosen for the watermark to be inserted because it contains high-frequency information that is less perceptible to the human visual system. This sub-band contains edges, textures, and other high-frequency details that are important for the optimum capacity, however, they are less noticeable to the human eye compared to the low-frequency sub-bands.

In this study, we used Haar wavelet as the mother wavelet that is a first-level DWT. To make the watermark less visible, we embedded it in the high-low sub-band of the wavelet decomposition.




Figure 3.3 a) Original image, b) 2D-DWT with 1-level decomposition

Following that, the high-low sub-band (a 256x256 image), which is the sub-band the watermark is placed inside, partitioned into 8x8 blocks. Also, we experimented with 16x16 and 32x32 blocks which correspond to watermark images of 16x16 and 8x8, respectively, to see the result. Then, each block undergoes an FrFT with transform orders $(a_1, a_2)$ that are subsequently utilized as encryption keys during the extraction

procedure. FrFTs and inverse FrFTs of the Lena image obtained using different transform can be seen in Figure 3.4.

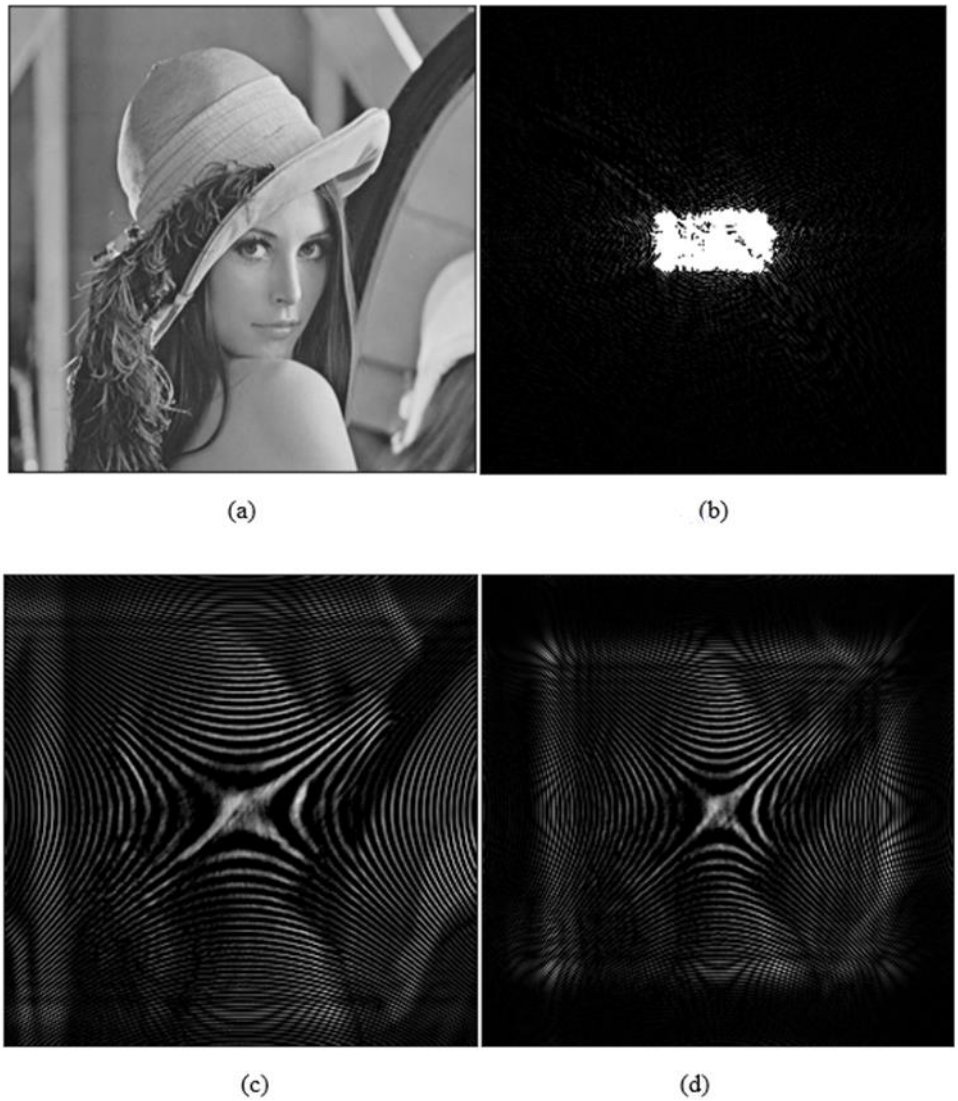



Figure 3.4 a) Lena image, b) magnitude FrFT of the Lena image with order pair (0.88, 1.06), c) inverse transformed image with order pair (-0.56,-1.55), d) inverse transformed image with order pair (-1.46,-0.45)

FrFT orders are employed as encryption keys to boost the resilience of the watermark in the face of various assaults. The use of encryption keys also provides security and prevents unauthorized access to the embedded watermark. Moreover, without the need for the original image a watermark can be obtained from the processed image even if it has been attacked rendering the suggested technique blind.

This implies that the watermark can be recovered even in the absence of original image.

In our suggested approach, a logo image is used as a watermark which is a 32x32 binary image consisting of 0's and 1's. It is important for the prosedure that the image needs to be binary because the embedding process modifying the FrFT coefficients of the high-low sub-band relies on the values of the pixels in the watermark image. If the watermark image is not binary, it must be first converted into a binary image to be properly placed in the original image.

To modify FrFT elements of the high-low sub-band image, the proposed method utilizes two pseudorandom noise (PN) sequences of size 1x64, which are generated based on standard normal (Gaussian) distribution. The initial step involves inspecting the first pixel situated at the top-left corner of the binary watermark image. blockThen it incorporated into the FrFT coefficients in the first 8x8 block of horizontal sub-band image. Conversely, if the pixel element is 0, the second PN sequence is amplified and incorporated. This procedure is reiterated for each binary-valued pixel in the 32x32 logo image, modifying all 1024 blocks in the horizontal sub-band image utilizing a specific PN sequence. As a result, the watermark data is integrated into the original image. The final step to apply is to revert to the spatial domain in this phase.

After altering the FrFT coefficients of horizontal sub-band image, watermark data is placed within the original image. To recover the processed image, we need to reverse the process by going back to the spatial domain. First, we apply the inverse FrFT (with appropriate order pair) to all 1024 modified 8x8 blocks to retrieve the modified high-low sub-band. In the end, merging the altered horizontal sub-band with the three untouched sub-bands and implementing an inverse DWT to get the watermarked image finalizes the process (see Figure 3.5).



Figure 3.5 Watermarked Lena image

### 3.2 Watermark Extracting Process

To extract the watermark, knowledge of the FrFT transform order pair and PN sequences is necessary, since they serve as encryption keys towards enhancing the security of the method. Procedure for watermark extraction phase can be visualized via the block diagram presented in Figure 3.6. Initially, the watermarked image undergoes a DWT to acquire the horizontal sub-band containing the watermark data, along with the other sub-bands that stay unchanged. The horizontal sub-band image is subsequently divided into 8x8 blocks and are subjected to a 2-D FrFT using the employed transform order pair. Next, the data samples within each 8x8 block are correlated separately with the two 1x64 PN sequences.

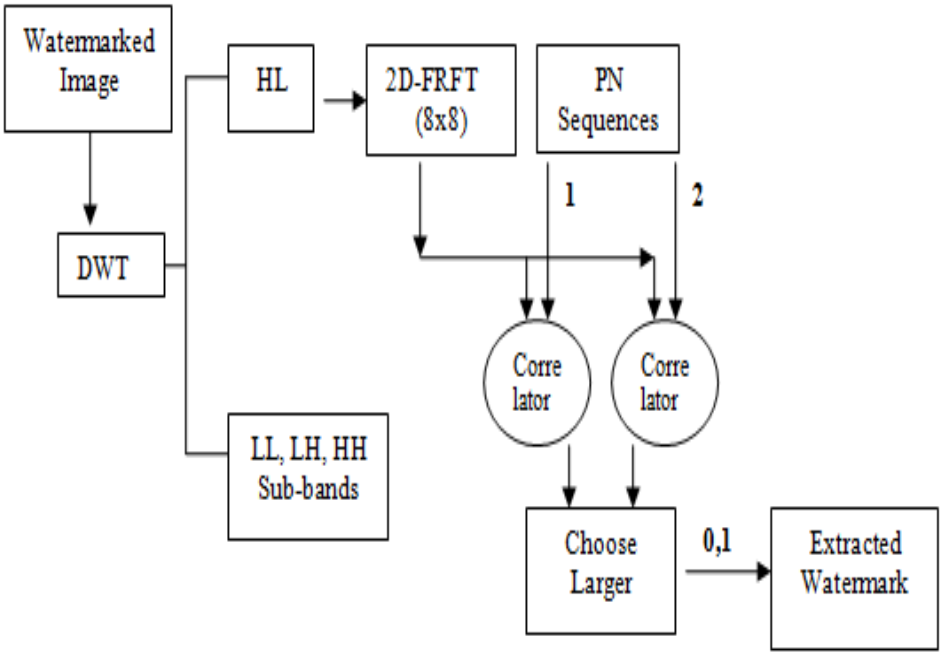
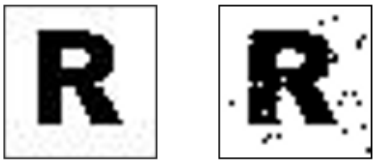



Figure 3.6 Flowchart representation of the method for extracting the watermark

Upon partitioning the horizontal sub-band image of watermarked image into 8x8 segments, each segment undergoes a 2-dimensional FrFT using the known transform order pair. The resulting FrFT coefficients of each block are then correlated with the two 1x64 PN sequences separately. Correlation values are compared, and the larger value is selected to match the pixel value. If the correlation value with the first PN sequence is greater, the pixel value is set to 1, otherwise, it is set to 0. This procedure is applied to all blocks in the horizontal sub-band image, resulting in the recovery of the 32x32 binary watermark image.



(a) (b)

Figure 3.7 a) Original watermark image b) Extracted watermark image

It is crucial to use the exact negative values of the transform order pair during the inverse FrFT process in the embedding process. Failure to use the correct transform orders may lead to inaccurate results. This is demonstrated in Figure 3.4.(a)-(d) using the Lena image. Figure 3.4.(b) displays the magnitude FrFT with order pair (0.88, 1.06). Performing the inverse FrFT with the correct orders (-0.88, -1.06) yields the original image. However, if incorrect orders such as (-0.56, -1.55) and (-1.46, -0.45) are used for the inverse FrFT, as shown in Figures 3.4.(c) and 3.4.(d), respectively, the original signal cannot be recovered. Therefore, it is essential to use the correct transform order pairs to ensure accurate extraction of the watermark.

To measure the efficacy of the suggested approach for watermark integration and retrieval, several experiments were conducted. The original Lena image (Figure 3.1.(a)) with a dimension of 512x512 was chosen as the test image for watermarking. A 32x32 binary image (Figure 3.1.(b)) was used as the watermark. As explained earlier, the high-low sub-band image in the DWT domain was selected as the embedding sub-band, as it provides better imperceptibility for the watermark. After dividing the high-low sub-band image into 8x8 blocks, the 2-D FrFT was applied to all the blocks using the FrFT order pair (1.21, 0.88). Based on the pixel values of the binary logo image, two PN sequences were multiplied and added into the appropriate blocks of DWT sub-band. Lena image that is watermarked (obtained using a gain factor of 2) is shown in Figure 3.5.

Next, we employed our proposed watermark extraction method using the watermarked image as input and followed the procedures illustrated in Figure 3.6. After applying all the necessary steps as explained earlier, we successfully retrieved the embedded watermark image, which is depicted in Figure 3.7.(b).

The excellence of a processed image, which contains 256 grayscale intensity values, can be evaluated using PSNR value. PSNR is a standard often employed to quantify the disparity between two images, where a higher PSNR value indicates minor changes between the initial and the distorted image. PSNR is expressed in decibels (dB) and is calculated based on the MSE between the two images (see Equations (2.1) and (2.2)).

In Equation (2.2), the mean squared error (MSE) between the NxN original, p(x,y), and the watermarked, p(x,y)', images are defined. Similarity between the original and extracted watermarks can be evaluated using the normalized correlation (NC) value, which is a measure of how much the extracted watermark resembles the original one (see Equation (2.3)). The NC measure extends from 0 to 1, with 1 signifying an impeccable correspondence between the original and extracted watermarks.

The watermarked image, retrieved using the proposed method with a gain factor of 2 as shown in Figure 3.7(b), achieved a PSNR value of 50.33 dB indicating high image quality. The extracted watermark image shown in Figure 3.7.(b) achieved an NC value of 0.92, signifying a substantial resemblance between the original and retrieved watermarks. It should be noted that a perfect recovery would result in an NC value of 1.

Instead of using PN sequences, another option for embedding the watermark is to use deterministic linear chirp signals. A linear chirp signal has a changing frequency over time and can be defined as a unit amplitude signal with initial frequency and sweep rate (see Equation (2.11) and Figure 2.4). This alternative approach is described in more detail in [9]. When sweep rate parameters of a chirp signal match the orders of the FrFT, it produces a compact impulse in that matching FrFT domain, due to the underlying basis functions of FrFTs with correct transform orders. This characteristic can be utilized for efficient extraction of the watermark. In our experiments, we also experimented with two chirp signals with different sweep rate parameters instead of PN sequences for watermark embedding. The sweep rate parameters can also serve as encryption keys for the watermark extraction process. During extraction, correlations are performed with the known chirp signals instead of PN sequences.

# CHAPTER FOUR
# EXPERIMENTAL RESULTS

To test the performance of the proposed watermark algorithm, we used two gray-scale images. One of them is Lena and the other one is Pepper image shown in Figure 4.1. To assess the resilience of our suggested watermark retrieval technique in the face of typical assaults, we conducted experiments with various types of attacks. For these tests, we set the constant gain factor, *c*, to 10, and we used different block sizes of 8x8, 16x16, and 32x32. We implemented additive white Gaussian noise with a variance of 0.005, and we also tried "jpeg" compressions with quality factors, Q, of 70%, 50%, and 30%. We further applied the cropping operation by cropping the central area of the images with the size of 32x32. In addition, we also experimented with two other types of attacks, namely histogram equalization and unsharp image. To measure the similarity between original and extracted images, we used NC metric. It is worth noting that the block sizes of 8x8, 16x16, and 32x32 correspond to PN sequences (or chirp signals) of length 64, 256, and 1024, respectively.



Figure 4.1 Original pepper image

## 4.1 Simulation Results

In the following tables, we present the PSNR and NC values of the attacked images that were watermarked using chirp signals and PN sequences with different block sizes. Furthermore, we used different gain factors, c, of chirp signals and blocks of 8x8 to retrieve the watermark images after "jpeg" compression attacks with quality factors of 80%, 70%, 60%, and 50%. The images of extraxted watermarks are shown in Table 4.1. When we look at the logos extracted after the attacks, we see that the results are quite acceptable and even with 50% JPEG compression, the logo can be easily distinguished by eye.

Table 4.1 Extracted watermark images with quality factors, Q, of a) 80%, b) 70%, c) 60%, d) 50%. Gain factor c = 8 is used with chirp signal

| **Attack on Image** | **Lena Image Extracted Watermark (*c* = 8)** | **Pepper Image Extracted Watermark (*c* = 8)** |
|---|---|---|
| "Jpeg" Compression (*Q* = 80%) |  |  |
| "Jpeg" Compression (*Q* = 70%) |  |  |
| "Jpeg" Compression (*Q* = 60%) |  |  |
| "Jpeg" Compression (*Q* = 50%) |  |  |

Table 4.2 PSNR values of Pepper watermarked images using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values

| **Constant gain** $C$ | **Chirp Signal 8x8 PSNR (dB)** | **Chirp Signal 16x16 PSNR (dB)** | **Chirp Signal 32x32 PSNR (dB)** | **PN Seqn. 8x8 PSNR (dB)** | **PN Seqn. 16x16 PSNR (dB)** | **PN Seqn. 32x32 PSNR (dB)** |
|---|---|---|---|---|---|---|
| 1 | 58.24 | 58.70 | 59.87 | 57.32 | 56.91 | 55.67 |
| 2 | 52.87 | 53.42 | 53.58 | 49.50 | 50.33 | 50.49 |
| 3 | 49.20 | 49.95 | 50.16 | 47.40 | 47.99 | 47.49 |
| 5 | 44.98 | 45.65 | 46.02 | 43.77 | 43.38 | 43.06 |
| 8 | 41.11 | 41.56 | 42.04 | 38.88 | 38.29 | 39.16 |
| 10 | 39.07 | 39.73 | 40.10 | 36.56 | 36.77 | 37.22 |
| 12 | 37.65 | 38.13 | 38.53 | 35.95 | 34.91 | 35.58 |
| 15 | 35.55 | 36.23 | 36.61 | 32.95 | 34.17 | 33.55 |
| 20 | 33.11 | 33.71 | 34.11 | 31.92 | 31.50 | 30.97 |
| 25 | 31.15 | 31.78 | 32.18 | 29.88 | 29.03 | 29.32 |
| 30 | 29.59 | 30.21 | 30.60 | 27.44 | 27.33 | 27.64 |
| 40 | 27.12 | 27.70 | 28.11 | 24.64 | 24.89 | 25.71 |

Table 4.3 PSNR values of Pepper watermarked images using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values

| **Constant gain *C*** | **Chirp Signal 8x8 PSNR (dB)** | **Chirp Signal 16x16 PSNR (dB)** | **Chirp Signal 32x32 PSNR (dB)** | **PN Seqn. 8x8 PSNR (dB)** | **PN Seqn. 16x16 PSNR (dB)** | **PN Seqn. 32x32 PSNR (dB)** |
|---|---|---|---|---|---|---|
| 1 | 58.24 | 58.70 | 59.87 | 57.32 | 56.91 | 55.67 |
| 2 | 52.87 | 53.42 | 53.58 | 49.50 | 50.33 | 50.49 |
| 3 | 49.20 | 49.95 | 50.16 | 47.40 | 47.99 | 47.49 |
| 5 | 44.98 | 45.65 | 46.02 | 43.77 | 43.38 | 43.06 |
| 8 | 41.11 | 41.56 | 42.04 | 38.88 | 38.29 | 39.16 |
| 10 | 39.07 | 39.73 | 40.10 | 36.56 | 36.77 | 37.22 |
| 12 | 37.65 | 38.13 | 38.53 | 35.95 | 34.91 | 35.58 |
| 15 | 35.55 | 36.23 | 36.61 | 32.95 | 34.17 | 33.55 |
| 20 | 33.11 | 33.71 | 34.11 | 31.92 | 31.50 | 30.97 |
| 25 | 31.15 | 31.78 | 32.18 | 29.88 | 29.03 | 29.32 |
| 30 | 29.59 | 30.21 | 30.60 | 27.44 | 27.33 | 27.64 |
| 40 | 27.12 | 27.70 | 28.11 | 24.64 | 24.89 | 25.71 |

We tested the method by changing gain factor value for both PN sequences and chirp signals. The perceptibility of the watermark is often dependent on a certain gain factor. The gain factor determines how strongly the watermark will be embedded into the image. The results are shown in Table 4.1.

The value of the gain factor is directly related to the visibility of the watermark on the image. The results in Table 4.2 and Table 4.3 shows that, when the gain factor is high, the watermark becomes more pronounced, but this can also lead to degradation in image quality. On the other hand, when the gain factor is very low, the watermark may not be strong enough and may be less resistant to attacks.

Therefore, the correct adjustment of the gain factor is important to optimize both the invisibility and the robustness of the watermark. In tables we see that even a gain factor of 20 is considered acceptable for PNSR values.

Table 4.4 PSNR values of Lena watermarked images, using PN sequences or chirp signals, different block sizes (8x8, 16x16, 32x32), and different gain factor values

| **Constant Gain $C$** | **Chirp Signal 8x8 PSNR (dB)** | **Chirp Signal 16x16 PSNR (dB)** | **Chirp Signal 32x32 PSNR (dB)** | **PN Seqn. 8x8 PSNR (dB)** | **PN Seqn. 16x16 PSNR (dB)** | **PN Seqn. 32x32 PSNR (dB)** |
|---|---|---|---|---|---|---|
| 1 | 58.05 | 58.70 | 59.87 | 55.27 | 55.67 | 56.09 |
| 2 | 52.87 | 53.42 | 53.58 | 50.82 | 50.62 | 50.63 |
| 3 | 49.20 | 49.95 | 50.16 | 47.68 | 47.59 | 47.17 |
| 5 | 44.98 | 45.65 | 46.02 | 42.64 | 42.85 | 43.23 |
| 8 | 41.11 | 41.56 | 42.04 | 39.12 | 38.80 | 39.07 |
| 10 | 39.08 | 39.73 | 40.11 | 37.31 | 36.54 | 37.17 |
| 12 | 37.65 | 38.13 | 38.54 | 36.96 | 35.42 | 35.59 |
| 15 | 35.56 | 36.24 | 36.62 | 33.21 | 34.16 | 33.78 |
| 20 | 33.13 | 33.72 | 34.14 | 31.16 | 30.97 | 31.03 |
| 25 | 31.19 | 31.80 | 32.23 | 28.55 | 29.47 | 29.28 |
| 30 | 29.65 | 30.25 | 30.67 | 28.52 | 27.81 | 27.56 |
| 40 | 27.20 | 27.77 | 28.21 | 25.23 | 25.16 | 25.07 |

In the following tables, we examined how the system tested in different images using same attacks. The technique shows the results of the same attacks on different images. As we can see from the tables, the results are independent of the images We also tested the system by dividing it into blocks of different sizes and we see that it is more resistant to attacks as the block sizes increase. But this increase could make it visually noticeable in the picture. It appears that the selection of block sizes needs to be done at an optimum level.

Table 4.5 NC values of Lena attacked images watermarked using chirp signals with different block sizes and with a constant gain factor $c = 2$

| **Attack on watermarked image** | **Chirp Signal 8x8 Normalized Correlation (NC)** | **Chirp Signal 16x16 Normalized Correlation (NC)** | **Chirp Signal 32x32 Normalized Correlation (NC)** |
|---|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ | 0.9667 | 0.9712 | 0.9897 |
| Cropping | 0.9962 | 0.9974 | 1.0000 |
| "Jpeg" Compression ($Q$ = 70%) | 0.9694 | 0.9948 | 1.0000 |
| "Jpeg" Compression ($Q$ = 50%) | 0.9686 | 0.9896 | 0.9897 |
| "Jpeg Compression" ($Q$ = 30%) | 0.9485 | 0.9818 | 0.9897 |
| Unsharp Image | 0.8622 | 0.9582 | 0.9897 |
| Histogram Equalization | 0.9949 | 0.9948 | 1.0000 |

Table 4.6 NC values of Pepper attacked images watermarked using PN sequences with different block sizes and with a constant gain factor $c = 10$

| **Attack on watermarked image** | **PN Sequence 8x8 Normalized Correlation (NC)** | **PN Sequence 16x16 Normalized Correlation (NC)** | **PN Sequence 32x32 Normalized Correlation (NC)** |
|---|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ | 0.9886 | 1 | 1 |
| Cropping | 0.9987 | 0.9479 | 0.9701 |
| "Jpeg" Compression ($Q$ =70%) | 0.9933 | 1 | 1 |
| "Jpeg" Compression ($Q$ = 50%) | 0.9618 | 0.9946 | 1 |
| "Jpeg" Compression ($Q$ = 30%) | 0.7686 | 0.9811 | 1 |
| Unsharp Image | 0.8753 | 0.9169 | 1 |
| Histogram Equalization | 0.9993 | 1 | 1 |

Table 4.7 NC values of Pepper attacked images watermarked using PN sequences with different block sizes and with a constant gain factor $c = 2$

| **Attack on watermarked image** | **PN Sequence 8x8 Normalized Correlation (NC)** | **PN Sequence 16x16 Normalized Correlation (NC)** | **PN Sequence 32x32 Normalized Correlation (NC)** |
|---|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ | 0.8002 | 0.8573 | 0.9895 |
| Cropping | 0.9695 | 0.9456 | 0.9701 |
| "Jpeg" Compression ($Q$ = 70%) | 0.7788 | 0.8330 | 0.9245 |
| "Jpeg" Compression ($Q$ = 50%) | 0.6768 | 0.7373 | 0.7777 |
| "Jpeg" Compression ($Q$ = 30%) | 0.6424 | 0.6470 | 0.7237 |
| Unsharp Image | 0.7265 | 0.6690 | 0.8429 |
| Histogram Equalization | 0.9608 | 0.9786 | 1 |

Table 4.8 C values of Lena attacked images watermarked using PN sequence signals with different block sizes and a constant gain factor $c = 2$

| **Attack on watermarked image** | **PN Sequence 8x8 Normalized Correlation (NC)** | **PN Sequence 16x16 Normalized Correlation (NC)** | **PN Sequence 32x32 Normalized Correlation (NC)** |
|---|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ | 0.9542 | 1.0000 | 1.0000 |
| Cropping | 0.9960 | 1.0000 | 1.0000 |
| "Jpeg" Compression ($Q$ = 70%) | 0.9688 | 0.9946 | 1.0000 |
| "Jpeg" Compression ($Q$ = 50%) | 0.9572 | 0.9865 | 1.0000 |
| "Jpeg" Compression ($Q$ = 30%) | 0.8518 | 0.9346 | 1.0000 |
| Unsharp Image | 0.8721 | 0.8310 | 1.0000 |
| Histogram Equalization | 0.9804 | 1.0000 | 1.0000 |

Table 4.9 NC values of Lena attacked images watermarked using PN sequence signals with different block sizes and with a constant gain factor $c = 2$

| Attack on watermarked image | PN Sequence 8x8 Normalized Correlation (NC) | PN Sequence 16x16 Normalized Correlation (NC) | PN Sequence 32x32 Normalized Correlation (NC) |
|---|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ | 0.7445 | 0.8160 | 0.9685 |
| Cropping | 0.9435 | 0.9654 | 1.0000 |
| "Jpeg" Compression ($Q$ = 70%) | 0.7385 | 0.8253 | 0.8539 |
| "Jpeg" Compression ($Q$ = 50%) | 0.6858 | 0.6867 | 0.7518 |
| "Jpeg" Compression ($Q$ = 30%) | 0.6466 | 0.5940 | 0.6889 |
| Unsharp Image | 0.6791 | 0.6416 | 0.6667 |
| Histogram Equalization | 0.9198 | 0.9616 | 1.0000 |

Table 4.10 Extracted watermarks of attacked images with constant gain factor $c = 5$

| **Attack on watermarked image** | **PN Sequence Extracted Watermark $c = 5$** | **Chirp Signal Extracted Watermark $c = 5$** |
|---|---|---|
| Additive WGN $\sigma^2 = 0.005$ |  |  |
| Cropping |  |  |
| "Jpeg" Compression ($Q = 70\%$) |  |  |
| "Jpeg" Compression ($Q = 50\%$) |  |  |
| "Jpeg" Compression ($Q = 30\%$) |  |  |
| Unsharp Image |  |  |
| Histogram Equalization |  |  |

# CHAPTER FOUR
# CONCLUSION

Watermarking techniques have become a focal point of interest in the current and preceding decades, driven largely by the swift evolution of technology and the escalating demand for transmitting extensive data online. The application of watermarking extends across diverse digital formats such as text, images, videos, and audio.

In the contemporary context, image watermarking emerges as a compelling subject of investigation, attracting fresh attention worldwide. This thesis examines a direct implementation of watermarking tailored to images, utilizing a binary image watermark code. The proposed method has undergone substantial testing on authentic image processing signal attacks. Moreover, the watermarking technique proposed in this thesis functions within the frequency domain, providing improved effectiveness and numerous benefits over alternative domains.

This thesis presents a secure technique for digital image watermarking, leveraging the fractional Fourier transform (FrFT) together with discrete wavelet transform (DWT). The proposed method delivers a resilient and secure watermarking algorithm, preserving the visual integrity of the watermarked image and facilitating the embedding and extraction of multiple watermarks from a sole original image. The significant contribution of FrFT lies in its security mechanism, utilizing transformation angles as secret keys in horizontal and vertical directions. Absence of these keys renders the extraction of the embedded watermark impossible, thereby enhancing the security of the watermarking process.

We tested techniques for embedding and extracting watermarks in the transform domain, utilizing a combination of DWT and FrFT. Notably, this prosess eliminates the need for original image during watermark extraction, rendering it a “blind” watermarking approach. To enhance security, we test the FrFT order and pseudo-noise (PN) sequences as encryption keys, fortifying the watermarking scheme against external threats. Experimental results affirm that this algorithm generates a watermark which is imperceptible to the human eye and resilient against a variety of image

processing attacks. To emulate real-world transmission scenarios of watermarked images, various combinations of attacks were applied to evaluate the algorithm's performance under composite attacks.

As observed in our experiments, there is almost no discernible difference between the pseudo-noise sequence and the chirp signal. The Peak Signal-to-Noise Ratio (PSNR) for watermarked Lena images indicate that when employing PN sequences or chirp signals, the use of 32x32 blocks yields slightly better results compared to 16x16 and 8x8 blocks. Thus, changing block size does not make a significant difference.

Based on the results of the conducted tests, it is evident that this algorithm and these methods can be applied in various scenarios and applications. They offer the flexibility to provide different levels of protection and can be enhanced using secret keys for added security.

## REFERENCES


Agullo O. E., M. I. Martin, & M. J. Castro (2005). FrFT-based image watermarking using a chirp signal. Proc. IEEE International Conference on Image Processing, vol. 2, pp. 834-837.

Almedia L. B. (November 1994). The fractional Fourier transform and time-frequency representations. IEEE Trans. Sig. Proc., vol. 42, pp. 3084 – 3091.

Alshebeili A., & M. A. Kutay (April, 2004). Robust watermarking of digital images using the fractional Fourier transform. Journal of Electronic Imaging, vol. 13, no. 2, pp. 295-304.

Anthony. G. Constantinides, & P. J. P. Linardos (2003). A review of image watermarking in the FrFT domain. Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 429-432.

Barni M., F. Bartolini, V. Capellini, & A. Piva (1998). A DCT-domain system for robust image watermarking. Signal Processing, vol. 66, no. 3, pp. 357–372.

Bultheel A., & H. Martinez-Sulbaran (May 2004). Computation of the fractional Fourier transform. Applied and Computational Harmonic Analysis, vol. 16, pp. 182-202.

Constantinides A. G., & P. J. P. Linardos (2003). "A review of image watermarking in the FrFT domain," in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 429-432

Cox I., J. Killian, F. T. Leighton, & T. Shamoon (December 1997). Secure spread spectrum watermarking for multimedia. IEEE Trans. on Image Proc, vol. 6, pp. 1673 – 1687.

Djurovic I., S. Stankovic, & I. Pitas (2001). Digital watermarking in the fractional Fourier transformation domain. Journal of Network and Computer Applications, vol. 24, pp. 167-173.

Dugad R., K. Ratakonda, & N. Ahuja (1998). A new wavelet-based scheme for watermarking images. Proceedings of 1998 International Conference on Image Processing (ICIP 1998), Chicago, IL, Vol. 2, 419-423.

Fen Z., M. U. Xiaomin, & Y. Shouyi (2005). Multiple chirp typed blind watermarking algorithm based on fractional Fourier transform. Proc. ISPACKS, pp. 141-144.

Gupta S., & K. Singh (2018). A review of discrete wavelet transform based digital image watermarking techniques. Multimedia Tools and Applications, 77(18).

Hong W.D., L.D. Ming, Y. Jun, & C.F. Xiong (2007). An improved chirp blind watermarking algorithm based on wavelet and fractional Fourier transform. Proc. IEEE International Conference on Images and Graphics, pp. 291-296.

Kundur D., & D. Hatzinakos (October 1997). A robust digital image watermarking method using wavelet-based fusion. Int. Conf. on Image Proc., vol. 3, pp. 544 – 547

Lei Gao, Lin Qi, Yongjin Wang, Enqing Chen, Shouyi Yang, & Ling Guan (December 2012). Rotation invariance in 2D-FRFT with application to digital image watermarking. J. Sign.Process. Syst. 72, pp. 133–148.

Mallat S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674-693.

Mohammed B., Ibrahim K., Tamer R., & Grigory K. Xu (November 2023). A robust DCT-based scheme for watermarking deep neural networks. The 10th International Symposium on Emerging Information, Communication and Networks. EICN 2023.

Mousavi S. M., A. Naghsh, & S. A. R. Abu Bakar (2014). Watermarking techniques used in medical Images: A survey. J. of Digit. Imaging, 27, 714–729.

Nweke N. N., A. M. Alhassan, & T. M. Agu (May 2013). Survey of image watermarking techniques. International Journal of Computer Applications, vol. 68, no. 9, pp. 36-43.

Nikolaidis N., & I. Pitas (28 May 1998). Robust image watermarking in the spatial domain. Signal Processing, Volume 66, Issue 3, Pages 385-403.

Ozaktas M., Z. Zalevsky, & M. A. Kutay (2001). The Fractional Fourier Transform, Wiley.

Zayed A. I., & M. A. Kutay (2021). Watermarking of digital images using the fractional Fourier transform. Proceedings of the International Conference on Image Processing, vol. 2, pp. 537-541.

Olcay Duman, & Olcay Akay (2011). A new method of wavelet domain watermark embedding and extraction using fractional Fourier transform. IEEE Proc. of 7th International Conference on Electrical and Electronics Engineering (ELECO), pp. 187-191.

Sathish R. S., & R. S. S. Venkatesan (2016). A study on image watermarking techniques using discrete wavelet transform. International Journal of Engineering and Technology, vol. 8, no. 6, pp. 2879-2883.

Solachidis V., & I. Pitas (November 2001). Circularly symmetric watermark embedding in 2D-DFT domain. IEEE Trans. on Image Proc., vol. 10, pp. 1741 – 1753.

Usha D. Nandini, & S. Divya (19–20 January 2017). A literature survey on various watermarking techniques. Proceedings of the 2017 International Conference on Inventive Systems and Control (ICISC), Coimbatore, India.

Vetterli M., & J. Kovacevic, (Eds.), (1995). Wavelets and Subband Coding (Vol. 2). Prentice Hall PTR.

Wang D., Li D., Yan J., & Chen F. (August 2007). An improved chirp typed blind watermarking algorithm based on wavelet and fractional Fourier transform. Fourth International Conference on Image and Graphics.

Yimeng Zhao, Chengyou Wang, Xiao Zhou, & Zhiliang Qin (2023). DARI-Mark: Deep learning and attention network for robust image watermarking. Mathematics 2023, 11, 209.

Yu F. Q., Z. K. Zang, & M. H. Xu (May 2006). A digital watermarking algorithm for image based on fractional Fourier transform. Proceedings of 1st IEEE Int. Conf. on Industrial Electronics and Appls., Singapore, pp. 1 – 5.

Zhang F., Mu X., & Yang S. (December 2005). Multiple-chirp typed blind watermarking algorithm based on fractional Fourier transform. Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems., Hong Kong.

## APPENDICES

### APPENDIX 1: Matlab Codes

```
clc

clear all

close all

%%%
```

```
f = (cH,[ones(1,32)*8],[ones(1,32)*8]);
 %%%% %%%% FRFT Of  CH Image Blocks %%%%%%% %%%%%%% %%
for i=1:32
   for j=1:32
   frft_image{i,j}=frft2d(f{i,j},[1.21 0.88]);
   end
end
 frac_image=frft_image;
%%%%%%%% Constant c=alfa %%%% %%%%%%%%%% %%%%%%%% %
 alfa=2;
%%%% %%%% Embedding Watermark To Image Process %%%%%%%%% %
%%%% %%%% Make Random Numbers Using Different Seeds %%%%%% %%
 %%%% %%%% Random Sequence and Keys %%%% %%%% %%%%%%%%
seed1 = random('norm',0,2000);
randn('seed',seed1);
key1 = randn(8,8);
seed2 = random('norm',0,2000);
randn('seed',seed2);
key2 = randn(8,8);
%%%% %%%% Embedding Watermark To Image %%%%%% %%%%% %%%%
 for m=1:32
   for n=1:32
```

```
        if watermark_image_binary(m,n)==0
        frac_image{m,n}=frac_image{m,n}+alfa*key1;
        else
        if watermark_image_binary(m,n)==1
        frac_image{m,n}=frac_image{m,n}+alfa*key2;
        end
        end
    end
end
 %%% %%%% Inverse FRFT of Image Blocks %%% %%%%%%%% %%%%
 for i=1:32
    for j=1:32
    inverse_frft{i,j}=frft2d(frac_image{i,j},[-1.21 -0.88]);
    end
end
 %%%%% %%%% Image Deblocks and Getting CH %%%%% %%%%%%% %
 inverse_frft_blocks=cell2mat(inverse_frft);
cH=real(inverse_frft_blocks);
 %%%% %%%%% IDWT Image %%%%% %%%%%%% %%%%%% %%%%
 watermarked_Image = idwt2(cA,cH,cV,cD,'haar');
 %%%% %%%% Show Watermarked Image %%%%%%%%%%%%%%%%%%
 image_watermarked=uint8(watermarked_Image);
```

```
figure,imshow(image_watermarked);

%%%%% %%%%%%% %%% PSNR VALUE %%%%% %%%%%%% %%%

PSNR(im2double(lena),im2double(image_watermarked))

%%%%%%% %%%%%%% %%%%%%%%% %%%%%%%%%% %%%% Attack on Image %%%% %%% %%%%% %%%%%%%%%% %%%%%%%%%% %%%%%%% %%%%%%%%%%

noisy_image = imnoise(image_watermarked,'gaussian',0,0.005);

[cropped_image] = centralcrop(image_watermarked,32,32);

[compress_image_70]=jpeg_comp(image_watermarked,80);

[compress_image_50]=jpeg_comp(image_watermarked,70);

[compress_image_30]=jpeg_comp(image_watermarked,60);

histogram_image=histeq(image_watermarked);

filt_image_median=median1(image_watermarked);

filt_image_wiener = wiener2(image_watermarked, [3 3]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%%%%% %%% Watermark Extraction %%%% %%%%%% %%%%%%%%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%% Wavelet of Image %%%%%%% %%%%% %%%%%%%%

[cA1,cH1,cV1,cD1] = dwt2(noisy_image,'haar');

[cA2,cH2,cV2,cD2] = dwt2(cropped_image,'haar');

[cA3,cH3,cV3,cD3] = dwt2(compress_image_70,'haar');

[cA4,cH4,cV4,cD4] = dwt2(compress_image_50,'haar');

[cA5,cH5,cV5,cD5] = dwt2(compress_image_30,'haar');
```

```
[cA6,cH6,cV6,cD6] = dwt2(histogram_image,'haar');

[cA7,cH7,cV7,cD7] = dwt2(filt_image_median,'haar');

[cA8,cH8,cV8,cD8] = dwt2(filt_image_wiener,'haar');

%[cA2,cH2,cV2,cD2] = dwt2(cD,'haar');

%%%%%%%% 8x8 Image Blocks %%%%%%%%%%%%% %%%%%%%%

f1 = mat2cell(cH1,[ones(1,32)*8],[ones(1,32)*8]);

f2 = mat2cell(cH2,[ones(1,32)*8],[ones(1,32)*8]);

f3 = mat2cell(cH3,[ones(1,32)*8],[ones(1,32)*8]);

f4 = mat2cell(cH4,[ones(1,32)*8],[ones(1,32)*8]);

f5 = mat2cell(cH5,[ones(1,32)*8],[ones(1,32)*8]);

f6 = mat2cell(cH6,[ones(1,32)*8],[ones(1,32)*8]);

f7 = mat2cell(cH7,[ones(1,32)*8],[ones(1,32)*8]);

f8 = mat2cell(cH8,[ones(1,32)*8],[ones(1,32)*8]);

%%%%%%%% FRFT Of Image Blocks %%%%%%% %%%%%% %%%%% %

 for i=1:32

    for j=1:32

    frac_image1{i,j}=frft2d(f1{i,j},[1.21 0.88]);

    frac_image2{i,j}=frft2d(f2{i,j},[1.21 0.88]);

    frac_image3{i,j}=frft2d(f3{i,j},[1.21 0.88]);

    frac_image4{i,j}=frft2d(f4{i,j},[1.21 0.88]);
```

```
    frac_image5{i,j}=frft2d(f5{i,j},[1.21 0.88]);
    frac_image6{i,j}=frft2d(f6{i,j},[1.21 0.88]);
    frac_image7{i,j}=frft2d(f7{i,j},[1.21 0.88]);
    frac_image8{i,j}=frft2d(f8{i,j},[1.21 0.88]);
    end
end

%%%%%% %%%%%%% Watermark Extraction %%%%% %%%%% %%%%
watermark1=[];
watermark2=[];
watermark3=[];
watermark4=[];
watermark5=[];
watermark6=[];
watermark7=[];
watermark8=[];

%%%%%%%%%%% Correlation of Images %%%%% %%%%% %%%%%
for i=1:32
    for j=1:32
        corr_zero1=corr2(key1,real(frac_image1{i,j}));
        corr_one1=corr2(key2,real(frac_image1{i,j}));
```

```
corr_zero2=corr2(key1,real(frac_image2{i,j}));
corr_one2=corr2(key2,real(frac_image2{i,j}));
corr_zero3=corr2(key1,real(frac_image3{i,j}));
corr_one3=corr2(key2,real(frac_image3{i,j}));
corr_zero4=corr2(key1,real(frac_image4{i,j}));
corr_one4=corr2(key2,real(frac_image4{i,j}));
corr_zero5=corr2(key1,real(frac_image5{i,j}));
corr_one5=corr2(key2,real(frac_image5{i,j}));
corr_zero6=corr2(key1,real(frac_image6{i,j}));
corr_one6=corr2(key2,real(frac_image6{i,j}));
corr_zero7=corr2(key1,real(frac_image7{i,j}));
corr_one7=corr2(key2,real(frac_image7{i,j}));
corr_zero8=corr2(key1,real(frac_image8{i,j}));
corr_one8=corr2(key2,real(frac_image8{i,j}));
      if corr_zero1>corr_one1
   watermark1(i,j)=0;
else
   watermark1(i,j)=1;
end
if corr_zero2>corr_one2
   watermark2(i,j)=0;
else
```

```
    watermark2(i,j)=1;
 end
 if corr_zero3>corr_one3
    watermark3(i,j)=0;
else
    watermark3(i,j)=1;
 end
 if corr_zero4>corr_one4
    watermark4(i,j)=0;
else
    watermark4(i,j)=1;
end
 if corr_zero5>corr_one5
    watermark5(i,j)=0;
else
    watermark5(i,j)=1;
end
if corr_zero6>corr_one6
    watermark6(i,j)=0;
else
    watermark6(i,j)=1;
end
```

```
        if corr_zero7>corr_one7
            watermark7(i,j)=0;
        else
            watermark7(i,j)=1;
        end
        if corr_zero8>corr_one8
            watermark8(i,j)=0;
        else
            watermark8(i,j)=1;
        end
    end
end
figure, imshow(uint8(255*watermark1))
figure, imshow(uint8(255*watermark2))
figure, imshow(uint8(255*watermark3))
figure, imshow(uint8(255*watermark4))
figure, imshow(uint8(255*watermark5))
figure, imshow(uint8(255*watermark8))
%figure, imshow(uint8(255*watermark7))
figure, imshow(uint8(255*watermark6))

%%%%%%%%%% Normalized Correlation Watermark and Extracted Image %%
```

```
R1 = correlation(watermark_image_binary,watermark1);

R2 = correlation(watermark_image_binary,watermark2);

R3 = correlation(watermark_image_binary,watermark3);

R4 = correlation(watermark_image_binary,watermark4);

R5 = correlation(watermark_image_binary,watermark5);

R6 = correlation(watermark_image_binary,watermark6);

R7 = correlation(watermark_image_binary,watermark7);

R8 = correlation(watermark_image_binary,watermark8);

    NC=[R1 R2 R3 R4 R5 R8 R6 ]


%%%
```

```
 %%%%%%%% Wavelet of Image Lena Image  %%%%%%%  %%%%%%%%
[cA,cH,cV,cD] = dwt2(lena,'haar');
 %%%%%%%%%%%%%%%% 8x8 Image Blocks of CH %%%%%% %%%%%
f = mat2cell(cH,[ones(1,32)*8],[ones(1,32)*8]);
%%%%%%%% FRFT Of  CH Image Blocks %%%%% %%%%% %%%%%%%
for i=1:32
   for j=1:32
   frft_image{i,j}=frft2d(f{i,j},[1.21 0.88]);
   end
end
 frac_image=frft_image;
 %%%%%%%% Constant c=alfa %%%%%% %%%%% %%%%%%%% %%%
          alfa=15;
 %%%%%%%% Embedding Watermark To Image Process %%%%%%%%%%
%%%%%%%% Make 2 chirps Using Different Sweep Rate %%%%%%%%%%
 for m=1:32
   for n=1:32
      if watermark_image_binary(m,n)==0
            chirp_1=makechirp(64,1,0.5);
            xx=frac_image{m,n};
            xx=(xx(:))';
            yy=(xx+alfa*(real(chirp_1)))';
```

```
            frac_image{m,n}=reshape(yy,8,8);
        else
            chirp_2=makechirp(64,1,0.75);
            xx=frac_image{m,n};
            xx=(xx(:))';
            yy=(xx+alfa*(real(chirp_2)))';
            frac_image{m,n}=reshape(yy,8,8);
        end
         end
end
 %%%%%%% Inverse FRFT of Image Blocks %%%%% %%%%%% %%%%%
 for i=1:32
    for j=1:32
    inverse_frft{i,j}=frft2d(frac_image{i,j},[-1.21 -0.88]);
    end
end
%%%%%%%%% Image Deblocks and Getting CH %%%%% %%%%%%%%
 inverse_frft_blocks=cell2mat(inverse_frft);
cH=real(inverse_frft_blocks);
%%%%%%%%% IDWT Image %%%%%%%%%%%%%%%%%%%%%% %
watermarked_Image = idwt2(cA,cH,cV,cD,'haar');
 %%%%%%%% Show Watermarked Image %%%%%%%%%%%% %%%%%
```

```
image_watermarked=uint8(watermarked_Image);
figure,imshow(image_watermarked);
%%%%%%%%%%%%%% PSNR VALUE %%%%%%%%%%%%%%%%
PSNR(im2double(lena),im2double(image_watermarked))
%%%%%%%%%% %%%%%%%%%%%% %%%%%%%%%%%%%%%% %%%%%%% %%%%%%% %%%%%%% Attack on Image %%%%%%% %%%%%%%%%%%%%%%%%% %%%%%% %%%% %%%%%% %%%%%
noisy_image = imnoise(image_watermarked,'gaussian',0,0.005);
cropped_image = centralcrop(image_watermarked,32,32);
cropped_image=uint8(cropped_image);
[compress_image_70]=jpeg_comp(image_watermarked,80);
[compress_image_50]=jpeg_comp(image_watermarked,70);
[compress_image_30]=jpeg_comp(image_watermarked,60);
histogram_image=histeq(image_watermarked);
filt_image_median=median1(image_watermarked);
filt_image_wiener = wiener2(image_watermarked, [3 3]);
%%%%%%%% %%%%%%%%% %%%%%%%%%%%% %%%%%% %%%%%%% %%% %%%%% %%%%%%% %%%%%% Watermark Extraction %%%%%%%%%%%%%%%%%
```

```
[cA4,cH4,cV4,cD4] = dwt2(compress_image_50,'haar');

[cA5,cH5,cV5,cD5] = dwt2(compress_image_30,'haar');

[cA6,cH6,cV6,cD6] = dwt2(histogram_image,'haar');

[cA7,cH7,cV7,cD7] = dwt2(filt_image_median,'haar');

[cA8,cH8,cV8,cD8] = dwt2(filt_image_wiener,'haar');

 %%%%%%%% 8x8 Image Blocks %%%%%%% %%%%%%% %%%%%%%% %%

 f1 = mat2cell(cH1,[ones(1,32)*8],[ones(1,32)*8]);

f2 = mat2cell(cH2,[ones(1,32)*8],[ones(1,32)*8]);

f3 = mat2cell(cH3,[ones(1,32)*8],[ones(1,32)*8]);

f4 = mat2cell(cH4,[ones(1,32)*8],[ones(1,32)*8]);

f5 = mat2cell(cH5,[ones(1,32)*8],[ones(1,32)*8]);

f6 = mat2cell(cH6,[ones(1,32)*8],[ones(1,32)*8]);

f7 = mat2cell(cH7,[ones(1,32)*8],[ones(1,32)*8]);

f8 = mat2cell(cH8,[ones(1,32)*8],[ones(1,32)*8]);

 %%%%%%%% FRFT Of Image Blocks %%%%%%%% %%%%%%%% %%%%

 for i=1:32

    for j=1:32

    frac_image1{i,j}=frft2d(f1{i,j},[1.21 0.88]);

    frac_image2{i,j}=frft2d(f2{i,j},[1.21 0.88]);

    frac_image3{i,j}=frft2d(f3{i,j},[1.21 0.88]);

    frac_image4{i,j}=frft2d(f4{i,j},[1.21 0.88]);

    frac_image5{i,j}=frft2d(f5{i,j},[1.21 0.88]);
```

```
    frac_image6{i,j}=frft2d(f6{i,j},[1.21 0.88]);
    frac_image7{i,j}=frft2d(f7{i,j},[1.21 0.88]);
    frac_image8{i,j}=frft2d(f8{i,j},[1.21 0.88]);
    end
end
 %%%%%%%%%%%% Watermark Extraction %%%%%%%%%%%%%%
watermark1=[];
watermark2=[];
watermark3=[];
watermark4=[];
watermark5=[];
watermark6=[];
watermark7=[];
watermark8=[];
 %%%%%%%%%%%% Correlation of Images %%%%%%%%%%%%%% %
 for i=1:32
    for j=1:32
        hh1=frac_image1{i,j};
        hh2=frac_image2{i,j};
        hh3=frac_image3{i,j};
        hh4=frac_image4{i,j};
        hh5=frac_image5{i,j};
```

```
hh6=frac_image6{i,j};

hh7=frac_image7{i,j};

hh8=frac_image8{i,j};

gg1=hh1(:);

gg2=hh2(:);

gg3=hh3(:);

gg4=hh4(:);

gg5=hh5(:);

gg6=hh6(:);

gg7=hh7(:);

gg8=hh8(:);


nn1=(corr2(real(gg1)',real(chirp_1)));

nn2=(corr2(real(gg2)',real(chirp_1)));

nn3=(corr2(real(gg3)',real(chirp_1)));

nn4=(corr2(real(gg4)',real(chirp_1)));

nn5=(corr2(real(gg5)',real(chirp_1)));

nn6=(corr2(real(gg6)',real(chirp_1)));

nn7=(corr2(real(gg7)',real(chirp_1)));

nn8=(corr2(real(gg8)',real(chirp_1)));

mm1=(corr2(real(gg1)',real(chirp_2)));

mm2=(corr2(real(gg2)',real(chirp_2)));
```

```
    mm3=(corr2(real(gg3)',real(chirp_2)));
    mm4=(corr2(real(gg4)',real(chirp_2)));
    mm5=(corr2(real(gg5)',real(chirp_2)));
    mm6=(corr2(real(gg6)',real(chirp_2)));
    mm7=(corr2(real(gg7)',real(chirp_2)));
    mm8=(corr2(real(gg8)',real(chirp_2)));

  if nn1>mm1  watermark1(i,j)=0;  else   watermark1(i,j)=1;   end
  if nn2>mm2  watermark2(i,j)=0;  else   watermark2(i,j)=1;   end
  if nn3>mm3  watermark3(i,j)=0;  else   watermark3(i,j)=1;   end
  if nn4>mm4  watermark4(i,j)=0;  else   watermark4(i,j)=1;   end
  if nn5>mm5  watermark5(i,j)=0;  else   watermark5(i,j)=1;   end
  if nn6>mm6  watermark6(i,j)=0;  else   watermark6(i,j)=1;   end
  if nn7>mm7  watermark7(i,j)=0;  else   watermark7(i,j)=1;   end
  if nn8>mm8  watermark8(i,j)=0;  else   watermark8(i,j)=1;   end
   end
   end
figure, imshow(uint8(255*watermark1))
figure, imshow(uint8(255*watermark2))
figure, imshow(uint8(255*watermark3))
figure, imshow(uint8(255*watermark4))
figure, imshow(uint8(255*watermark5))
```

```
figure, imshow(uint8(255*watermark6))

figure, imshow(uint8(255*watermark7))

figure, imshow(uint8(255*watermark8))

imwrite((uint8(255*watermark1)),'C:\Users\Olcay\Documents\MATLAB\8x8 Chirp\watermark1.png');

%%%%% %%%%% Normalized Correlation Watermark and Extracted Image %%

 R1 = correlation(watermark_image_binary,watermark1);

 R2 = correlation(watermark_image_binary,watermark2);

 R3 = correlation(watermark_image_binary,watermark3);

 R4 = correlation(watermark_image_binary,watermark4);

 R5 = correlation(watermark_image_binary,watermark5);

 R6 = correlation(watermark_image_binary,watermark6);

 R7 = correlation(watermark_image_binary,watermark7);

 R8 = correlation(watermark_image_binary,watermark8);

 C=[R1 R2 R3 R4 R5 R8 R6 ];
```