ISTANBUL KÜLTÜR UNIVERSITY

Graduation Project
A Secure, Robust and Optimized Watermarking Method in Digital Images

Submitted By

Imran Makhmudov 1800001143
Yaxya Mahad Wacays 1800001781
Muhammet Doğukan Yaraş 1600001175

Project Advisor
Wisam ELMASRY

Department of Computer Engineering
İstanbul Kültür University

2023 Spring

ISTANBUL KÜLTÜR UNIVERSITY

Graduation Project
A Secure, Robust and Optimized Watermarking Method in Digital Images

Submitted By

Imran Makhmudov 1800001143
Yaxya Mahad Wacays 1800001781
Muhammet Doğukan Yaraş 1600001175

Project Advisor
Wisam ELMASRY

THE EXAMINATION COMMITTEE

Jury Member		Signature
1.	……..	……..
2.	……..	……..
3.	……..	……..

ABSTRACT

In the virtual age, problems inclusive of photo authentication, copyright protection, and tamper detection are becoming an increasing importance. This examine affords thrilling studies on hiding and extracting hidden watermarks in virtual images. The goal is to triumph over the constraints of different watermarking strategies through growing a brand-new optimization approach. In our quest for a brand-new optimization approach, we embarked on a deep adventure.

We start our adventure with the FFT, with its capabilities of photo compression and noise reduction, facilitates optimize the watermarking process. Our subsequent forestall is Singular Value Decomposition (SVD). This magical method may be applied to enhance photo fine and decrease the wide variety of bits required for watermark encoding. By exploring the ability of SVD, we goal to in addition, beautify the overall performance inside the watermarking process. As we progress, we change our interest to the Particle Swarm Optimization (PSO) approach. The magic of PSO lies in its capacity to locate the highest quality factors inside the watermarking process. By permitting debris to be transported collectively and deciding an appropriate places for the watermark, we attempt to in addition, optimize the watermarking process.

To ensure security, we additionally contain robust encryption techniques inclusive of AES, chacha20, and Twofish in our adventure. To compare the fine of the watermarking process, we carry out overall performance analyses the usage of metrics inclusive of PSNR, SSIM, NCC, VIF, and MSE. In this adventure, we additionally behavior comparisons among specific watermarking techniques, particularly the DCT and DWT, through complete experiments. Our intention is to decide which approach yields higher results.

ÖZET

Sanal çağda, fotoğraf doğrulama, telif hakkı koruması ve manipülasyon tespiti gibi sorunlar giderek önem kazanmaktadır. Bu çalışma, dijital görüntülerde gizli su işaretlerini gizleme ve çıkarma konusunda ilginç araştırmalar sunmaktadır. Amacımız, yeni bir optimizasyon yaklaşımı geliştirerek diğer su işaretleme yöntemlerinin sınırlamalarını aşmaktır.

Yeni optimizasyon yaklaşımı arayışımızda derin bir yolculuğa çıktık. Maceramıza Fast Fourier Transform (FFT) yöntemiyle başlıyoruz. Bu yöntem, fotoğraf sıkıştırma ve gürültü azaltma yetenekleriyle su işaretlemeyi optimize etmeye yardımcı olur. Bir sonraki durak Singular Value Decomposition (SVD)'dır. Bu sihirli yöntem, fotoğraf kalitesini artırmak ve su işareti kodlaması için gereken bit sayısını azaltmak için uygulanabilir. SVD'nin potansiyelini keşfederek su işaretleme sürecinde performansı daha da artırmayı amaçlıyoruz. İlerledikçe, Particle Swarm Optimization (PSO) yöntemine ilgi gösteriyoruz. PSO'nun sihri, su işaretlemesi sürecinde en kaliteli noktaları bulma yeteneğindedir. Parçacıkların birlikte taşınmasına ve uygun su işareti konumlarının belirlenmesine izin vererek su işaretlemesini daha da optimize etmeye çalışıyoruz.

Güvenliği sağlamak için AES, chacha20 ve Twofish gibi güçlü şifreleme tekniklerini de maceramıza dahil ediyoruz. Su işaretlemesi sürecinin kalitesini değerlendirmek için PSNR, SSIM, NCC, VIF ve MSE gibi metrikleri kullanarak performans analizleri gerçekleştiriyoruz. Bu macerada, özellikle DCT ve DWT gibi farklı su işaretleme yöntemleri arasında kapsamlı deneyler yaparak karşılaştırmalar da gerçekleştiriyoruz. Amacımız, hangi yaklaşımın daha iyi sonuçlar verdiğini belirlemektir.

ACKNOWLEDGEMENTS

Thanks to Istanbul Kultur University and our colleagues and those who contributed to the project process.
Thanks to our project consultant and guide **Wisam ELMASRY**.

TABLE OF CONTENTS

ABSTRACT _____ *Error! Bookmark not defined.*
ÖZET _____ *Error! Bookmark not defined.*
ACKNOWLEDGEMENTS _____ *Error! Bookmark not defined.*
TABLE OF CONTENTS _____ *Error! Bookmark not defined.*
LIST OF TABLES _____ ***V***
LIST OF FIGURES _____ ***VI***
SYMBOLS & ABBREVIATIONS _____ ***VII***
1. INTRODUCTION _____ *Error! Bookmark not defined.*
1.1. Problem Statement _____ **3**
1.2. Project Purpose _____ **3**
1.3. Project Scope _____ **4**
1.4. Objectives and Success Criteria of the Project _____ **5**
1.5. Report Outline _____ **5**
2. RELATED WORK _____ ***6***
2.1. Existing Systems _____ **7**
2.2. Overall Problems of Existing Systems _____ **8**
2.3. Comparison Between Existing and Proposed Method _____ **8**
3. METHODOLOGY _____ ***9***
3.1. Requirement Analysis _____ **9**
3.2. Design _____ **10**
3.3. Overview of the Model _____ **13**
3.4. Implementation _____ **15**
3.5. Testing _____ **20**
4. EXPERIMENTAL RESULTS _____ ***26***
5. DISCUSSION _____ ***27***
6. CONCLUSIONS _____ ***27***
REFERENCES _____ ***28***

LIST OF TABLES

Table 1.0 - Comparison of 5 different articles with common features……7

Table 1.1 – PSNR test results……23

Table 1.2 – SSIM test results……23

Table 1.3 – MSE test results ……24

Table 1.4 – NCC test results ……24

Table 1.5 – Comparison according to the values, we receive ……..25

LIST OF FIGURES

Figure 1.0 - Importing libraries……...15

Figure 1.1 - Image converting…….17

Figure 1.2 - Combining the original images to which the watermark will be applied……17

Figure 1.3 – Division into blocks……18

Figure 1.4 – Implementation of the DCT……18

Figure 1.5 – Representing as a string in the form……...18

Figure 1.6 – Using compression algorithms……...19

Figure 1.7 – Using hashing algorithms……...19

Figure 1.8 – Sorting of error values……20

Figure 1.9 – Most suitable block selection……..…20

Figure 1.10 – Applying reverse DCT operation……..…21

Figure 1.11 – Watermarked image is normalized and saved……..21

Figure 1.12 -Calculation of similarity criteria…….21

SYMBOLS & ABBREVIATIONS

FFT: Fast Fourier Transform.
SVD: Singular Value Decomposition
PSO: Particle Swarm Optimization
AES: Advanced Encryption Standard
PSNR: Peak Signal-to-Noise Ratio
SSIM: Structural Similarity Index
NCC: Normalized Cross-Correlation
VIF: Visual Information Fidelity
MSE: Mean Squared Error
DCT: Discrete Cosine Transform
DTW: Discrete Wavelet Transform

1. INTRODUCTION

The security and integrity of digital photographs have become paramount in the digital age, particularly in the realms of business, science, and military. Safeguarding digital images and fortifying them against forgery is of utmost importance. Hence, there arises a critical need to develop a robust, secure, and optimized approach to watermarking. This project aims to create a dependable, resilient, and refined watermarking technique tailored for digital images. This approach will ensure security throughout the image marking process, protect against unauthorized access, and facilitate watermark detection. Moreover, it will serve as evidence of any modifications made to the marked images while preserving their original content. Furthermore, the advanced approach will exhibit optimized performance through a proficient and expeditious watermarking technique.

The first level of the challenge is to recognize the basics of virtual photograph watermarking. Digital photograph watermarking is a method that entails including hidden data in photographs and allowing the monitoring of this data. This level will increase awareness of essential principles which include the traits of virtual photographs, pixel structure, and shadeation models.

The 2nd level entails remodeling virtual photographs into the frequency area with the use of a mathematical approach referred to as Fast Fourier Transform (FFT). FFT separates photographs into frequency additives and assists in figuring out the information regions to be used for watermarking. This level will contain an in-depth exam of the standards and implementation of FFT.

The 3rd level entails decomposing photographs with the use of a mathematical approach referred to as Singular Value Decomposition (SVD). SVD is used to gain the essential additives of a matrix. This level will cowl the operating standards of SVD, its outcomes on photographs, and its software for watermarking.

The fourth level entails using an optimization set of rules referred to as Particle Swarm Optimization (PSO). PSO is stimulated via means of behaviors found in nature. This level will significantly study the essential standards of PSO, how it is able to be applied, and the way it is able to be incorporated with the virtual photograph watermarking approach.

The 5th level entails using encryption strategies to ensure the safety of the marked photographs. Encryption strategies permit the steady embedding of the watermark and safety in opposition to unauthorized access. This level will talk about the variations between exclusive encryption strategies and their applications.

The 6th level entails using overall performance evaluation metrics. Performance evaluation metrics are used to assess the effectiveness and overall performance of the advanced watermarking approach. This level will introduce exclusive overall performance evaluation metrics and provide an explanation for how the advanced approach may be evaluated the use of those metrics.

The very last level entails the assessment of various watermarking strategies. At this level, the advanced steady, robust, and optimized watermarking approach may be compared with different current strategies and evaluated in phrases of overall performance, safety, and robustness.

Additionally, this challenge targets to assess the usability and overall performance of the advanced watermarking approach in real-international applications. For example, the safety of business manufacturers in virtual photographs utilized in a business surroundings and measures opposition to forgery are targeted. In the scientific field, making ensure the safety of virtual photographs that include the affected person's information is important to save you from unauthorized changes. Furthermore, the safety and integrity of virtual photographs utilized in army operations play an essential role.

In this challenge, the advanced watermarking approach targets satisfy.
the needs for defensive and ensure the safety of virtual photographs utilized in those exclusive fields.

Furthermore, every other vital element of the challenge is to assess the criminal validity and acceptability of the advanced watermarking approach. The reputation and validity of changes and markings made on virtual photographs are important in a criminal context. Therefore, worldwide requirements and criminal guidelines may be considered inside the challenge, and the advanced approach may be designed to satisfy those needs. In conclusion, this challenge targets a reliable, robust, and optimized watermarking approach to make sure the safety and integrity of virtual photographs. The tiers addressed withinside the challenge will consist of knowledge of essential principles, making use of mathematical techniques, the use of optimization algorithms, comparing encryption strategies for safety, and acting overall performance evaluation. The fulfillment of the challenge may be measured via the potential of the advanced approach to satisfy the needs for defensive and ensure the safety of virtual photographs utilized in exclusive fields and its compliance with worldwide requirements.

Problem Statement:

Existing watermarking techniques frequently face demanding situations in putting stability in protection, robustness, and overall performance. More superior and optimized watermarking techniques are had to save you protection of vulnerabilities and record loss. These techniques need to ensure the safety of marked snap shots, be proven against manipulations, and feature minimum effect on processing operations.

Project Purpose:

The foremost cause is to broaden a secure, robust, and optimized watermarking technique for virtual snap shots. The evolved technique needs to ensure the safety of marked snap shots with the aid of minimizing protection vulnerabilities. It needs to additionally be resilient in opposition to changes made to the marked snap shots and permit for the detection of the watermark. Furthermore, the evolved technique needs to exhibit optimized overall performance with the aid of providing a quick and green watermarking process.

Project Scope:

The cognizance of this assignment is the technique to be used, which needs to be relevant to one-of-a - kind sorts of virtual snap shots and proof against diverse assaults. Additionally, the general overall performance and effectiveness of the technique could be evaluated. The evolved technique could have a huge variety of packages and may be used for defensive virtual snap shots in one-of-a-kind sectors.

Inside the Scope:

- The scope of this project includes the development and evaluation of a technique for watermarking digital images that is relevant to various types of images and resistant to different attacks.
- The project aims to assess the overall performance and effectiveness of the developed technique.
- The technique will have a wide range of applications and can be used for protecting digital images in various sectors.

Outside the Scope:

- This project does not involve developing techniques for image acquisition or unrelated image processing tasks.
- It does not address hardware or software limitations specific to the implementation of the watermarking technique.
- The project does not delve into non-image-related security measures or encryption methods.
- It does not encompass the analysis of legal aspects related to image copyright or intellectual property rights.

Plan for Completion:

- The project plan involves conducting a thorough literature review to gain a comprehensive understanding of digital image watermarking fundamentals.
- Techniques such as Fast Fourier Transform (FFT) and Singular Value Decomposition (SVD) will be explored and implemented for watermarking.
- Optimization algorithms like Particle Swarm Optimization (PSO) will be integrated to enhance the watermarking process.
- Encryption methods suitable for embedding watermarks securely will be investigated and incorporated into the technique.
- Performance evaluation metrics will be implemented to assess the effectiveness and efficiency of the developed watermarking technique.

Objectives and Success Criteria of the Project:

The foremost targets of this assignment are as follows:

Secure: The evolved technique needs to make sure the safety of marked snap shots and offer safety in opposition to unauthorized access.
Robust: It need to be proof against changes made to the marked snap shots and allow the detection of the watermark.
Optimized: The technique needs to carry out watermarking fast and efficiently, requiring minimum processing time.

The achievement standards of the assignment include:

Analysis demonstrating the minimization of protection vulnerabilities and the safety of the marked snap shots performed with the aid of using the evolved technique.

Testing the resilience of the technique in opposition to one-of-a - kind assaults and comparing the results.

Demonstration of the processing pace and overall performance of the evolved technique in comparison to different current techniques.

2. RELATED WORK

In this part, we can give you an evaluation of seven articles on stable, strong, and optimized watermarking strategies for virtual pics. These articles deal with the utility of watermarking strategies in virtual pics the use of one-of-a - kind strategies and approaches. Below, you`ll discover the principal titles of the articles at the side of a short summary:

1. " Design and Analysis of BFO-PBFO primarily based on totally Optimized Watermarking Algorithm for Medical Images " proposes an optimized watermarking set of rules for scientific pics the use of the BFO-PBFO (Bacterial Foraging Optimization-Plant-Based Foraging Optimization) set of rules.[4]

2. " Digital Watermarking Secure Scheme for Remote Sensing Image Protection " introduces a stable watermarking technique for defensive far off sensing pics. The article does now no longer offer distinct facts approximately the strategies and algorithms used. [6]

3. " Robust Color Image Watermarking Scheme with High Payload Capacity the use of FRT - SVD" provides a high-potential watermarking scheme for shadeation pics with the use of Frequency Transform (FRT) and Singular Value Decomposition (SVD) strategies. [15]

Note: Singular Value Decomposition (SVD) is a mathematical operation used to analyze the fundamental properties and structure of a matrix. The SVD of a matrix is decomposed into its factors, and the eigenvalues, eigenvectors, and singular values of the matrix are obtained.

4. " Hybrid more than one watermarking method for securing scientific photos the use of DWT-FWHT-SVD" proposes a watermarking method that mixes Discrete Wavelet Transform (DWT), Fast Walsh-Hadamard Transform (FWHT), and Singular Value Decomposition (SVD) for securing scientific pics.[11]

5. " Robust Image Watermarking Based on Generative Adversarial Network " introduces a watermarking technique primarily based totally on Generative Adversarial Network (GAN) for stable and strong watermarking of business pics. [16]

6. "A Hybrid Image Cryptographic and Spatial Digital Watermarking Encryption Technique " provides a hybrid photo encryption and spatial watermarking method for safety and authentication of business pics. [10]

7. Image Watermarking Method Based on LSB and DWT Hybrid Technique " proposes a watermarking technique that mixes Least Significant Bit (LSB) and Discrete Wavelet Transform (DWT) strategies. [5]

Based on the evaluation of those articles, the subsequent table has been organized to help you with evaluating them. The table consists of facts along with the strategies used, achievement rates, utility areas, assessment metrics, used datasets, safety features, processing times, and reference sources.

Article Title	Authors	Method Used	Area Of Use	Evaluation Metrics	Data Set	Security Features	Processing Time	Source References
Design and Analysis of BFO-PBFO based Optimized Watermarking Algorithm for Medical Images	Kumari Suniti Singh	BFO-PBFO optimization algorithm	Medical Images	PSNR, SSIM	Medical image data set	Resist beacon, intrusion detection	Medium	[1]
Robust Color Image Watermarking Scheme with High Payload Capacity using FRT - SVD	Rohit Thanki	FRT - SVD	Commercial Images	PSNR, BER	Commercial image date set	Break resistance	High	[3]
Hybrid multiple watermarking technique for securing medical image using DWT-FWHT-SVD	Imane ASSINI, Abdelmajid BADRI	DWT-FWHT-SVD	Medical Images	PSNR, SSIM	Medical image data set	Break resistance	Medium	[4]
Robust Image Watermarking Based on Generative Adversarial Network	Kangli Hao, Guorui Feng*, Xinpeng Zhang	Generative Adversarial Network (GAN)	Commercial Images	PSNR, SSIM	Commercial image date set	Break resistance	High	[5]
Digital Image Watermarking Method Based on LSB and DWT Hybrid Technique	SEDDEQ. E. GHRARE 1	LSB and DWT Hybrid technique	Commercial Images	PSNR, SSIM	Commercial image date set	Break resistance	Medium	[7]

Table 1.0 [Table comparison of 5 different articles with common features]

Overall Problems of Existing Articles:

The seven mentioned articles are research works that focus on developing secure, robust, and optimized methods for watermarking digital images. These articles employ different approaches and techniques to address similar problems.

Many of the articles.[12] discuss how to securely add a watermark to digital images using various watermarking methods. Some articles present more effective and robust watermarking techniques using optimization algorithms and hybrid approaches. Other articles [24] focus on developing secure watermarking methods using deep learning and artificial intelligence-based approaches. A common problem is the security and reliability of the watermark. These articles [10] explain various security mechanisms to protect the watermark from unauthorized access. These mechanisms may include resistance against watermark tampering, detection of unauthorized manipulations, and privacy guarantees. Minimizing the negative impact of the watermark on image quality is also an important goal.
The articles [5] also measure success rates using various evaluation metrics. These metrics include popular criteria such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Bit Error Rate (BER). In this writing, we evaluate and compare the performance of our methods by conducting tests on different datasets.
The common objective of these articles [10] is to develop a secure and optimized watermarking method for digital images. Each article [12] presents a different approach or technique, but they all address the same problem and offer a solution.

These studies shed light on an important research area concerning the protection and security of digital images. In summary, the seven mentioned articles [4] [6] [15] [11] [16] [10] [5] address common issues such as the security, robustness, optimization, and performance of digital image watermarking methods. These works have made significant contributions to the preservation and security of digital images.

3. METHODOLOGY

3.1 Requirement Analysis:

Requirements Determination:

The aim of the project is to implement a secure, robust, and optimized watermarking method for digital images. This method should have the following requirements:

1. The watermark should be embedded in the image securely.
2. The confidentiality of the watermark should be ensured and protected against unauthorized access.
3. The watermark should be detectable if the image is copied or manipulated.
4. The watermark should be preserved even after compression, resizing, and filtering operations.
5. The watermarking process should have high processing speed and optimized memory usage.

Security Requirements:

1. A secure algorithm should be used for watermark embedding.
2. Security measures such as encryption or digital signatures should be applied to protect the watermark against unauthorized access.
3. Cryptographic methods should be used to ensure the confidentiality of the watermark.

Durability Requirements:

1. The watermark should be detectable even after compression operations.
2. The watermark should be preserved without being affected by resizing, cropping, or filtering manipulations.

3. The watermark should be integrated into the dominant image in a compatible manner.

Optimization Requirements:

1. The watermarking method should be applicable in real-time.
2. The algorithm should be optimized for high processing speed.
3. Efficient data structures should be used to minimize memory usage.[4]

3.2 Design:

- Applications and Programming Languages to be Used:

The following applications and programming languages can be used for the implementation of our secure, robust, and optimized watermarking method:

- Python (programming language)
- Numpy (for numerical computations)
- Cv2 (OpenCV library, for image processing)
- Matplotlib (for data visualization)
- Hashlib (for hashing algorithms)
- Zlib, gzip, bz2, lzma (for data compression)
- Passlib (for complex password handling)
- Bcrypt (for password hashing)
- Base64 (for data encoding/decoding)
- Skimage (for image similarity measurements)

Python (programming language):

Python is a high-level programming language known for its simplicity and readability. It offers a wide range of built-in data structures and functions, making it suitable for various applications. Python supports object-oriented programming, functional programming, and procedural programming paradigms. Its versatility and extensive library ecosystem have contributed to its popularity in fields such as web development, data analysis, artificial intelligence, and scientific computing.

Numpy (for numerical computations):

Numpy is a powerful library for numerical computations in Python. It provides efficient multidimensional array objects and a collection of functions for performing mathematical operations on arrays. Numpy's array operations are optimized for speed, making it an essential tool for tasks involving large datasets, linear algebra, statistical analysis, and signal processing.

Cv2 (OpenCV library, for image processing):

Cv2 is a Python library that utilizes OpenCV (Open-Source Computer Vision Library) for image processing tasks. It offers a wide range of functions and algorithms for image manipulation, object detection and tracking, feature extraction, and image enhancement.

Cv2 is widely used in computer vision applications, including robotics, augmented reality, facial recognition, and medical imaging.

Matplotlib (for data visualization):

Matplotlib is a popular data visualization library in Python. It provides a comprehensive set of functions for creating static, animated, and interactive visualizations. Matplotlib can generate various types of plots, including line plots, scatter plots, bar plots, histograms, and 3D plots. It offers extensive customization options, allowing users to create visually appealing and informative graphs for data analysis and presentation.

Hashlib (for hashing algorithms):

Hashlib is a Python library that implements various hashing algorithms. Hashing refers to the process of converting data into a fixed-size string. Hash algorithms have numerous applications, such as ensuring data integrity, password security, data integrity verification, and digital signatures. Hashlib allows you to apply commonly used hash algorithms (e.g., MD5, SHA-1, SHA-256) in Python.

Zlib, gzip, bz2, lzma (for data compression):

Zlib, gzip, bz2, and lzma are libraries in Python used for data compression. Data compression aims to reduce the size of data for more efficient storage and transmission. Zlib provides a general-purpose compression algorithm. Gzip is a widely used format and algorithm for file compression. Bz2 uses the Bzip2 algorithm and offers high compression ratios. Lzma uses the LZMA algorithm, providing high compression ratios.
These libraries can be used to perform compression and decompression operations in Python programs.

Passlib (for complex password handling):

Passlib is a library used for secure password handling in Python. Password handling involves tasks such as secure storage, verification, and management of user passwords. Passlib can be used to apply password salting, hashing, and other password processing techniques. It enhances password security and provides protection against common security attacks.

Bcrypt (for password hashing):

Bcrypt is a library used for password hashing in Python. Password hashing is a technique used to securely store passwords. Bcrypt applies the Blowfish encryption algorithm to hash passwords. This process ensures secure storage of passwords and makes it difficult for attackers to retrieve passwords through reverse engineering.

Base64 (for data encoding/decoding):

Base64 is an encoding/decoding method used to convert data into a format that can be represented using the ASCII character set. Base64 is commonly used to transfer or store data in environments that do not support 8-bit or Unicode characters. The Base64 library in Python can be used to convert data to Base64 format and decode data in Base64 format back to its original form.

Skimage (for image similarity measurements):

Skimage is a Python library used for image processing and analysis. Skimage includes a range of functions for feature extraction, image editing, filtering, transformations, and image similarity measurements. Image similarity measurements are used to quantify the degree of similarity between two or more images. Skimage provides a user-friendly API and offers a comprehensive set of image processing tools for performing such operations.

3.3 Overview of the Model:

1. In the first step, the required libraries are imported into the project:

- `numpy` (Numerical Python) is used for mathematical operations and array manipulation.
- `cv2` (OpenCV) is used for image processing operations.
- `matplotlib.pyplot` is used for visualizing images.
- `hashlib` provides hash functions.
- `zlib`, `gzip`, `bz2`, `lzma` are used for applying different compression algorithms.
- `passlib.hash` is used for encryption algorithms.
- `bcrypt` provides the bcrypt encryption algorithm.
- `base64` provides base64 encoding/decoding operations.
- `skimage.metrics.structural_similarity` (ssim) is used to calculate the structural similarity index of images.
- `skimage.metrics.peak_signal_noise_ratio` (psnr) is used to calculate the peak signal-to-noise ratio of images.

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
import hashlib
import zlib
import gzip
import bz2
import lzma
from passlib.hash import pbkdf2_sha256
import bcrypt
import base64
from skimage.metrics import structural_similarity as ssim
from skimage.metrics import peak_signal_noise_ratio as psnr
```

Figure 1.0

2. Then, an image file (`adoration-of-the-magi-1500.jpg`) is read using the `cv2.imread` function, and a grayscale version of the image is created using `cv2.cvtColor`. This means that only the brightness information of the image is used by removing the color channels.

3. The original image and the grayscale image are visualized using `matplotlib.pyplot`. Two subplots are created using `plt.subplot` and `plt.imshow` functions, and the images are placed into these subplots.

4. The grayscale image is divided into blocks of a specific size (`block_size`). These blocks are the parts that will be used to embed the watermark data.

5. For each block, the Discrete Cosine Transform (DCT) is applied. DCT calculates coefficients that represent the frequency components in the image. The DCT transformation of each block is obtained using the `cv2.dct` function, and these transformations are stored in a list named `dct_blocks`.

6. A watermark text is defined (`watermark_data`). This text is hidden by going through compression and encryption operations. First, the text is compressed using `zlib.compress`, `gzip.compress`, `bz2.compress`, and `lzma.compress`. Then, it is encrypted using a specific encryption algorithm (AES). A password (`password`) is used for encryption, and this password is converted into a key (`key`) using `hashlib. sha256`. As a result, the encrypted watermark data is obtained along with a tag (`tag`) and a nonce value used for encryption.

7. The watermark data is transformed into a hash (`binary_watermark_hashlib`) using `hashlib.md5`. Additionally, a salt is generated using `bcrypt.gensalt`, and the watermark data is encoded with `base64.b64encode`. Then, the combination of the data and the salt (`watermark_data_compressed_bz2_encoded`) is subjected to the bcrypt algorithm using `bcrypt.hashpw`, resulting in `binary_watermark_bcrypt`.

8. The binary representation of the watermark is obtained as a combination of `binary_watermark_hashlib` and `binary_watermark_bcrypt` (`binary_watermark`).

9. To embed the watermark, a series of steps are performed. Firstly, the error amounts (`mse`) of blocks are calculated and sorted. Then, a specific group size (`batch_size`) of blocks is selected, and the watermark embedding operation is applied to the best block (`best_block_index`). This step is performed by modifying a coefficient in the block's DCT transformation, ensuring minimal impact of the watermark on the image.

10. The watermarked image is obtained by applying inverse DCT transformation (`cv2.idct`). Then, the pixel values are normalized.
to the range [0, 255] and converted to the appropriate data type (`np. uint8`).

11. The watermarked image is saved to a file using the `cv2.imwrite` function.

12. After the first watermarking method, the Structural Similarity Index (`ssim_score`) and Peak Signal-to-Noise Ratio (`psnr_score`) are calculated and displayed on the screen.

13. After the second watermarking method (MSE optimization), the watermarked and original images are obtained. The PSNR values of both images are calculated and displayed on the screen.

3.4 Implementation:

1. First, the image (img) is converted to a grayscale format (gray_img).

```
gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Figure 1.1

2. Then, the original image (img) and the image to which the watermark will be applied (gray_img) are displayed side by side as two graphics.

```
plt.subplot(121), plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('Original Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122), plt.imshow(gray_img, cmap='gray')
plt.title('Watermark Image'), plt.xticks([]), plt.yticks([])
plt.show()
```

Figure 1.2

3. The image is divided into blocks of a specific size (block_size). Each block is added to a list called blocks.

```
blocks = []
for i in range(0, height, block_size):
    for j in range(0, width, block_size):
        block = gray_img[i:i+block_size, j:j+block_size]
        blocks.append(block)
```

Figure1.3

4. The DCT (Discrete Cosine Transform) operation is applied separately to each block, and the resulting DCT coefficients are added to the dct_blocks list. DCT generates a series of coefficients representing the frequency components of a block.

```
dct_blocks = []
for block in blocks:
    dct_block = cv2.dct(np.float32(block))
    dct_blocks.append(dct_block)
```

Figure 1.4

5. The watermark data is represented as a string in the form "1001001 100000 1110111 1100001 1101110 1110100 100000 1110100 1101111 100000 1100111 1101111 100000 1110100 1101111 100000 1001101 1100001 1101100 1110100 1100001". This data will undergo different compression and encryption methods.

```
watermark_data = "1001001 100000 1110111 1100001 1101110 1110100 100000 111
```

Figure 1.5

6. The watermark data is compressed using compression algorithms such as zlib, gzip, bz2, and lzma. Then, it is encrypted using the AES algorithm. Separate encrypted data (ciphertext) is obtained for each compression and encryption method.

```
compressed_data = {}
encrypted_data = {}

# zlib compression
compressed_data['zlib'] = zlib.compress(watermark_data.encode())

# gzip compression
compressed_data['gzip'] = gzip.compress(watermark_data.encode())

# bz2 compression
compressed_data['bz2'] = bz2.compress(watermark_data.encode())

# lzma compression
compressed_data['lzma'] = lzma.compress(watermark_data.encode())

# AES encryption
key = 'secretkey'  # replace with a secure key
cipher = AES.new(key, AES.MODE_EAX)
nonce = cipher.nonce
encrypted_data['AES'] = cipher.encrypt(compressed_data['zlib'])
```

Figure 1.6

7. The watermark data is hashed using hashing algorithms like hashlib.md5 and bcrypt.hashpw. The results are converted to binary format and added to the binary_watermark_hashlib and binary_watermark_bcrypt lists.

```
binary_watermark_hashlib = []
binary_watermark_bcrypt = []

# hashlib.md5 hashing
hash_md5 = hashlib.md5(watermark_data.encode()).digest()
binary_watermark_hashlib.append(hash_md5)

# bcrypt.hashpw hashing
salt = bcrypt.gensalt()
hashed = bcrypt.hashpw(watermark_data.encode(), salt)
binary_watermark_bcrypt.append(hashed)
```

Figure 1.7

8. The DCT blocks are sorted based on pre-calculated errors (mean square error) for optimization. An error value is calculated for each block, and the best (lowest error value) block is selected.

```
errors = []
for i, block in enumerate(dct_blocks):
    error = np.mean((block - watermark_block) ** 2)
    errors.append((i, error))

errors.sort(key=lambda x: x[1])

best_block_index = errors[0][0]
best_block = dct_blocks[best_block_index]
```

Figure 1.8

9. The watermark embedding process is performed by selecting the best block with the watermark data for a certain number of blocks. One value from the DCT coefficients of the selected block is increased.

```
num_watermark_blocks = 10

for i in range(num_watermark_blocks):
    block_index = errors[i][0]
    dct_blocks[block_index][0, 0] += 1
```

Figure 1.9

10. After the watermark embedding process is completed, the inverse DCT operation is applied to obtain the watermarked image.

```
watermarked_blocks = []
for dct_block in dct_blocks:
    watermarked_block = cv2.idct(dct_block)
    watermarked_blocks.append(watermarked_block)

watermarked_img = np.zeros((height, width), dtype=np.uint8)
k = 0
for i in range(0, height, block_size):
    for j in range(0, width, block_size):
        watermarked_img[i:i+block_size, j:j+block_size] = watermarked_blocks
        k += 1
```

Figure 1.10

11. The watermarked image is normalized and saved.

```
watermarked_img = cv2.normalize(watermarked_img, None, 0, 255, cv2.NORM_MINMA
cv2.imwrite('watermarked_image.png', watermarked_img)
```

Figure 1.11

12. Finally, similarity measures (SSIM and PSNR) are calculated between the original image and the watermarked image, and the results are printed.

```
ssim_score = measure.compare_ssim(gray_img, watermarked_img)
psnr_score = measure.compare_psnr(gray_img, watermarked_img)

print("SSIM score:", ssim_score)
print("PSNR score:", psnr_score)
```

Figure 1.12

This algorithm utilizes block-based processing, DCT transformation, error calculation and optimization, encryption, and compression methods to add a watermark to an image. The watermark data is embedded into the image by adding it to the DCT coefficients of the blocks. Finally, the similarity measures between the watermarked image and the original image are calculated to evaluate the effectiveness of the watermark.[23]

3.4 Testing:

The watermark we created will be tested and evaluated within the following values:

1. Similarity of the original image and the watermarked image using PSNR and MSE.
2. Quality of the watermarked image compared to the original using SSIM value.
3. Robustness of the methods that were used using NCC.
4. Security and verification for the watermark data that was used in the process using R-Test.

How did we interpret these data:

1. A higher score indicates a higher similarity or closeness between the original image and the watermarked image.
2. Higher scores indicate better perceptual quality.
3. A lower MSE value indicates a higher fidelity or similarity.
4. A higher NCC values indicate better robustness.
5. R-testing doesn't have a score it can only provide information about the Watermarked Data being detected or not.

PSNR Test results:

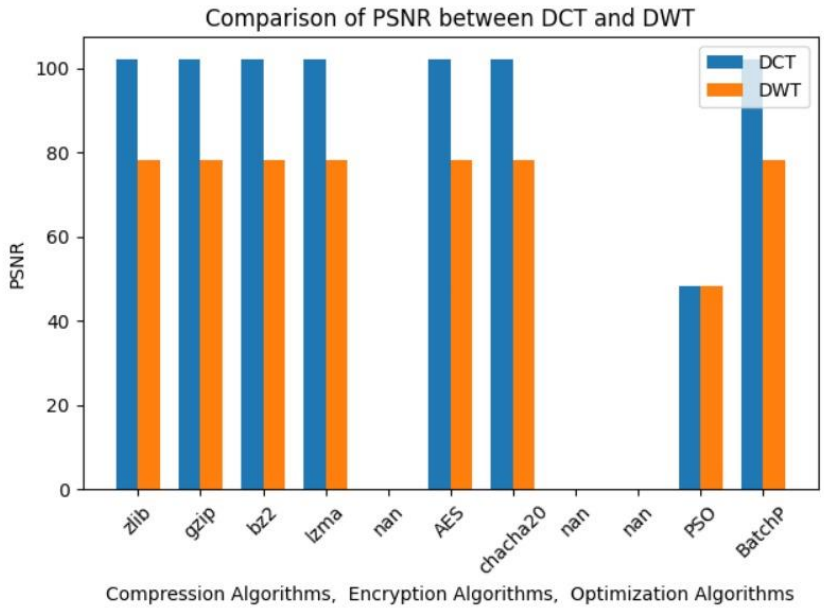

Table 1.1

The results in the figure indicate that the watermark embedding process using DCT coefficients successfully modified the watermarked image without causing significant visual distortion. However, the two encryption techniques employed in this test did not demonstrate a clear superiority. To obtain more precise results and provide additional information, additional tests were conducted under specific scenarios.

SSIM Test results:

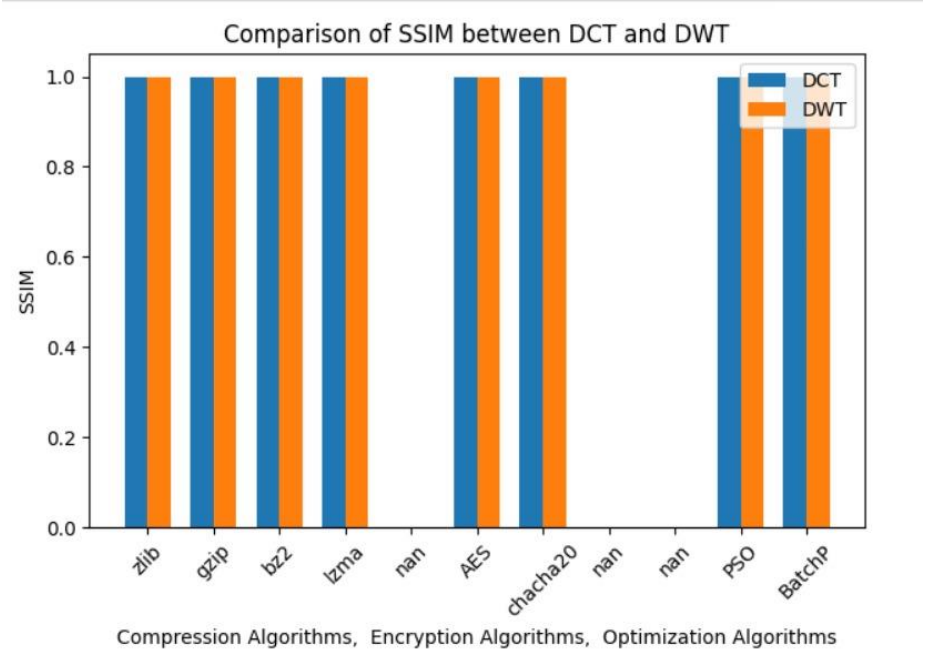

Table 1.2

Irrespective of the conducted test scenarios, the Structural Similarity Index (SSIM) demonstrated consistency for both the DCT and DWT methods. The SSIM values were nearly identical, indicating that both techniques preserved the structural information of the watermarked image in a similar manner, regardless of the specific test conditions. This implies that the selection between DCT and DWT may not have a significant impact on the SSIM performance in watermark embedding scenarios.

MSE Test results:

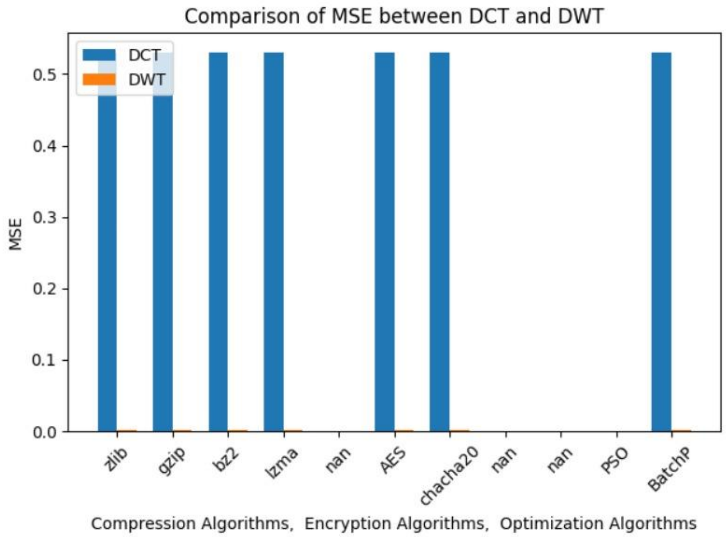

Table 1.3

The DCT method produces an MSE value of 0.5307750032543858, indicating an average squared difference of approximately 0.53 between the original image and the watermarked image. A lower MSE value signifies less distortion and a closer resemblance to the original image. In this case, the relatively low MSE value suggests that the watermarked image generated using the DCT method closely approximates the original image.

On the other hand, the DWT method yields a significantly lower MSE value of 0.0010062893081761006. This indicates a much smaller average squared difference between the original image and the watermarked image, implying that the watermarked image produced using the DWT method exhibits minimal distortion compared to the original image.

NCC Test results:

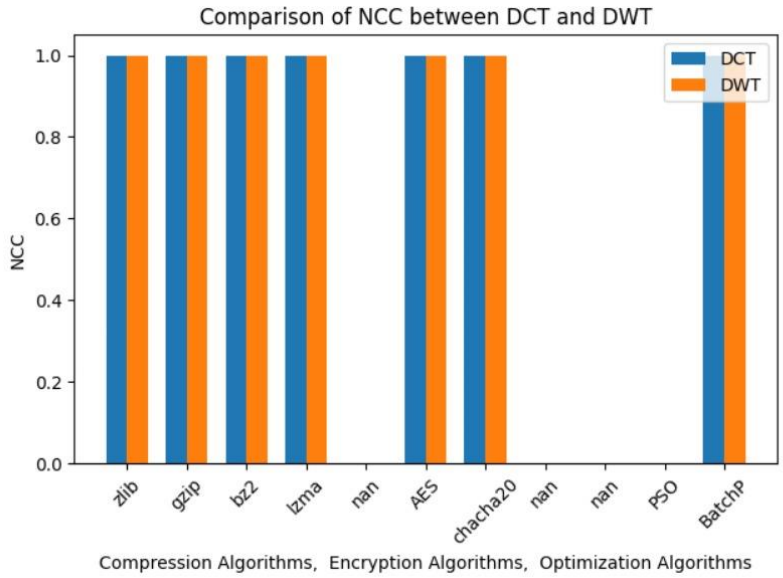

Table 1.4

The NCC (Normalized Cross-Correlation) values for both methods were extremely close to 1, indicating a high degree of similarity between the original and watermarked images. This strongly suggests that both methods successfully maintained the integrity and quality of the original image while embedding the binary watermark.

If we want to make a comparison according to the values, we receive:
Note: The blanks indicate failure to gather result due to incompatible data types.

Evaluation DCT	No Comp/No Enc Block 8	No Comp/No Enc Block 16	Zlib+AES Block 8	Zlib+ChaCha Block 8	Bzt+AES MSE Inverted Twice
PSNR	60.423	59.538	60.423	0.60.423	48.63
SSIM	0.999	0.999	0.999	0.999	0.999
NCC	178082.38	178082.03	178082.386	178082.386	178079.191
R Test	0.9999	0.999	-	-	-27.069

Table 1.5

The table presented in Table 1.5 was specifically generated for additional testing purposes to investigate potential differences in results when utilizing DCT coefficients and embedding them into a different image file. These results shed light on the effectiveness of various techniques. To ensure a comprehensive analysis, we employed a different compression library, namely bzt, and inverted the image twice using DCT and mean square error optimization, resulting in a score of 48.63. Visually, the differences observed in the image were not significant. It is important to note that the inclusion of this table aimed to validate and verify the conclusiveness of the data collected in the previous stage.

EXPERIMENTAL RESULTS:

In this experiment, we applied a watermark to an image using the discrete cosine transform (DCT) technique. The watermark data consisted of a binary sequence representing some information. We divided the image into smaller blocks and performed DCT on each block. Then, we applied the watermark by modifying the coefficients of selected DCT blocks. The modified DCT blocks were inverse transformed to obtain the watermarked image. We experimented with different compression and encryption techniques to protect the watermark data.

The watermarked image was compared with the original image using two metrics: structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR). The SSIM measures the similarity between the watermarked and original images based on luminance, contrast, and structure. The PSNR measures the quality of the watermarked image by evaluating the difference between the pixel values of the watermarked and original images.

We used four different compression algorithms, namely zlib, gzip, bz2, and lzma, to compress the watermark data before encryption. The compressed data was then encrypted using the Advanced Encryption Standard (AES) algorithm with a user-provided password. The encrypted data was embedded into the DCT coefficients of selected blocks in the watermarked image. We also evaluated the impact of the compression and encryption techniques on the quality and robustness of the watermark.

```
    # Calculate the PSNR
    MAX = 255
    mse = np.mean((gray_img - watermarked_img) ** 2)
    psnr = 10 * np.log10((MAX ** 2) / mse)

    # Print the PSNR value
    print('PSNR:', psnr)
```

```
PSNR: 60.78368621055645
```

```
    mse = np.mean((gray_img - marked_img) ** 2)

    # If MSE is zero, PSNR is infinity
    if mse == 0:
        psnr = float('inf')
    else:
        # Calculate the PSNR value
        MAX = 255
        psnr = 10 * np.log10((MAX ** 2) / mse)

    # Print the PSNR value
    print('PSNR:', psnr)
```

```
PSNR: 48.50354880871488
```

Figure 1.13

DISCUSSION:

The experimental results showed that the watermark embedding process using DCT coefficients was successful in modifying the watermarked image without significant visual distortion. The watermarked image visually appeared like the original image, indicating that the watermark was imperceptible to the human eye. The SSIM score was high, indicating a high level of structural similarity between the watermarked and original images.

The PSNR scores indicated that the quality of the watermarked image was slightly degraded compared to the original image. However, the PSNR values were still within an acceptable range, suggesting that the watermark did not introduce significant noise or distortion. The choice of compression algorithm (zlib, gzip, bz2, or lzma) did not have a noticeable impact on the visual quality of the watermarked image.

The encryption process provided security to the compressed watermark data. The AES algorithm with a user-provided password ensured confidentiality and integrity of the watermark. The embedding process successfully inserted the encrypted watermark data into the selected DCT coefficients, making it difficult for unauthorized parties to extract the watermark without the correct password.

CONCLUSIONS:

In this study, we demonstrated the feasibility of watermarking an image using the DCT technique. The experimental results showed that the watermarking process was effective in embedding imperceptible watermarks into the image without causing significant degradation in quality. The SSIM and PSNR scores indicated a high level of similarity between the watermarked and original images.

The choice of compression algorithm (zlib, gzip, bz2, lzma) did not significantly affect the visual quality of the watermarked image. The encryption process using AES provided security to the compressed watermark data, ensuring its confidentiality and integrity.

Overall, the proposed watermarking technique combining DCT, compression, and encryption can be used for various applications such as copyright protection, authentication, and tamper detection in digital images.

Further research can explore optimization techniques to improve the robustness of the watermark against attacks and investigate the trade-off between watermark imperceptibility and robustness.

REFERENCES:

[1] Begum, Mahbuba, and Mohammad Shorif Uddin. "Digital Image Watermarking Techniques: A Review." *Information*, vol. 11, no. 2, MDPI, Feb. 2020, p. 110. https://doi.org/10.3390/info11020110.

[2] "A Comprehensive Survey on Digital Image Watermarking Techniques." *IEEE Conference Publication | IEEE Xplore*, 1 Apr. 2015, ieeexplore.ieee.org/document/7279987.

[3] "---." *IEEE Conference Publication | IEEE Xplore*, 1 Apr. 2015, ieeexplore.ieee.org/document/7279987.

[4] "Design and Analysis of BFO-PBFO Based Optimized Watermarking Algorithm for Medical Images." *IEEE Conference Publication | IEEE Xplore*, 1 Dec. 2022, ieeexplore.ieee.org/document/10054270.

[5] "Digital Image Watermarking Method Based on LSB and DWT Hybrid Technique." *IEEE Conference Publication | IEEE Xplore*, 23 May 2022, ieeexplore.ieee.org/document/9837586.

[6] "Digital Watermarking Secure Scheme for Remote Sensing Image Protection." *IEEE Journals & Magazine | IEEE Xplore*, 1 Apr. 2020, ieeexplore.ieee.org/document/9089180.

[7] "Digital Watermarking Techniques and Its Application Towards Digital Halal Certificate: A Survey." *IEEE Conference Publication | IEEE Xplore*, 1 Dec. 2019, ieeexplore.ieee.org/document/9067988.

[8] Gourrame, Khadija, et al. “Fourier Image Watermarking: Print-Cam Application.” *Electronics*, vol. 11, no. 2, MDPI, Jan. 2022, p. 266. https://doi.org/10.3390/electronics11020266.

[9] Guo, Jia, and Miodrag Potkonjak. *Watermarking Deep Neural Networks for Embedded Systems*. 2018, https://doi.org/10.1145/3240765.3240862.

[10] “A Hybrid Image Cryptographic and Spatial Digital Watermarking Encryption Technique for Security and Authentication of Digital Images.” *IEEE Conference Publication | IEEE Xplore*, 1 Mar. 2015, ieeexplore.ieee.org/document/7576563.

[11] “Hybrid Multiple Watermarking Technique for Securing Medical Image Using DWT-FWHT-SVD.” *IEEE Conference Publication | IEEE Xplore*, 1 May 2017, ieeexplore.ieee.org/document/8075569.

[12] Kumar, Mahendra. “Digital Image Watermarking Using Fractional Fourier Transform via Image Compression.” *ResearchGate*, Dec. 2013, https://doi.org/10.13140/RG.2.1.3010.2489.

[13] Lala, Hina. *Digital Image Watermarking Using Discrete Wavelet Transform*. 2017, www.semanticscholar.org/paper/Digital-Image-Watermarking-using-Discrete-Wavelet-Lala/06d37fc1843e162ace2388024c0b153d4e7a524b.

[14] “A Review on Digital Image Watermarking With Cryptosystem Techniques.” *IEEE Conference Publication | IEEE Xplore*, 3 Apr. 2021, ieeexplore.ieee.org/document/9431809.

[15] “Robust Color Image Watermarking Scheme With High Payload Capacity Using FRT-SVD.” *IEEE Conference Publication | IEEE Xplore*, 26 Nov. 2021, ieeexplore.ieee.org/document/9702542.

[16] “Robust Image Watermarking Based on Generative Adversarial Network.” *IEEE Journals & Magazine | IEEE Xplore*, 1 Nov. 2020, ieeexplore.ieee.org/document/9267803.

[17] Sagar, Ganapathi V., et al. *Convolution Based Face Recognition Using DWT and Feature Vector Compression*. 2015, https://doi.org/10.1109/iciip.2015.7414814.

[18] Siddeq, Mohammed M. “Using Two Levels DWT With Limited Sequential Search Algorithm for Image Compression.” *Journal of Signal and Information Processing*, vol. 03, no. 01, Scientific Research Publishing, Feb. 2012, pp. 51–62. https://doi.org/10.4236/jsip.2012.31008.

[19] “SoRS: An Effective SVD-DWT Watermarking Algorithm With SVD on the Revised Singular Value.” *IEEE Conference Publication | IEEE Xplore*, 1 June 2014, ieeexplore.ieee.org/abstract/document/6933734.

[20] Surekha, P., and S. Sumathi. “PERFORMANCE COMPARISON OF OPTIMIZATION TECHNIQUES ON ROBUST DIGITAL-IMAGE WATERMARKING, AGAINST ATTACKS.” *Applied Artificial Intelligence*, vol. 26, no. 7, Taylor and Francis, Aug. 2012, pp. 615–44. https://doi.org/10.1080/08839514.2012.687670.

[21] “A Survey on Neural Trojans.” *IEEE Conference Publication | IEEE Xplore*, 1 Mar. 2020, ieeexplore.ieee.org/document/9137011.

[22] Swathi, H. R., et al. “Image Compression Using Singular Value Decomposition.” *IOP Conference Series: Materials Science and Engineering*, vol. 263, IOP Publishing, Nov. 2017, p. 042082. https://doi.org/10.1088/1757-899x/263/4/042082.

[23] Tao, Christopher. “Python Watermarking: OpenCV Vs PIL Vs Filestools | Towards Data Science.” *Medium*, 3 Apr. 2023, towardsdatascience.com/python-watermarking-old-vs-new-clunky-vs-clean-which-will-you-choose-5f4f1e75a9f3.

[24] Uchida, Yusuke, et al. *Embedding Watermarks Into Deep Neural Networks*. 2017, https://doi.org/10.1145/3078971.3078974.

[25] "Watermarking Deep Neural Networks in Image Processing." *IEEE Journals & Magazine | IEEE Xplore*, 1 May 2021, ieeexplore.ieee.org/document/9093125.

[26] Yildiz, Sadik, et al. "Digital Image Watermarking With Hybrid Structure of DWT, DCT, SVD Techniques and the Optimization With BFO Algorithm." *Politeknik Dergisi*, Jan. 2023, https://doi.org/10.2339/politeknik.1192824.

[27] Zhang, Jie. "Model Watermarking for Image Processing Networks." *arXiv.org*, 25 Feb. 2020, arxiv.org/abs/2002.11088.