Advanced Techniques in Digital Image Watermarking and Custom SVD Implementation

Emre Safa Yalçın
150210090

Computer Engineering, Istanbul Technical University

May 12, 2024

Abstract

This paper explores the integration of a custom implementation of Singular Value Decomposition (SVD) and digital watermarking techniques to enhance digital media security. We develop and evaluate robust watermarking algorithms that utilize custom SVD for embedding and extracting watermarks without perceptible quality loss, assessed by the Peak Signal-to-Noise Ratio (PSNR).

Keywords: digital watermarking, SVD, image processing, PSNR, security

1 Introduction

The proliferation of digital media and the need for copyright protection have driven the development of digital watermarking techniques. These methods need to be robust, transparent, and resilient to various attacks. Traditional watermarking techniques often rely on built-in library functions for operations such as Singular Value Decomposition (SVD), which are not always optimized for the specific requirements of watermarking. This paper introduces a custom implementation of SVD designed to enhance the security and efficiency of digital watermarking systems.

2 Singular Value Decomposition Implementation

2.1 Overview of SVD

Singular Value Decomposition (SVD) is a matrix decomposition technique used in many signal processing and statistics applications. It decomposes any given matrix into three other matrices (U, Sigma, and Vt), where Sigma contains the singular values and U and Vt contain the left and right singular vectors, respectively. This decomposition is pivotal for applications like image compression and noise reduction.

2.2 Custom SVD Algorithm

Our SVD implementation is crafted from scratch using Python, which specifically avoids reliance on high-level functions from libraries such as NumPy or SciPy for core computations. The core of our implementation is the eigenvalue and eigenvector calculation using a custom QR decomposition algorithm.

```
def qr_decomposition(A):
    # Implementation details that factorize A into Q and R matrices.
    return Q, R

def qr_algorithm(A, iterations=1000, tol=1e-10):
    # Use QR decomposition iteratively to find eigenvalues and eigenvectors
    return eigenvalues

def eigenvectors_from_eigenvalues(A, eigenvalues)
    #Use null_space function to find the null space of A - lambda*I
    return eigenvectors

def svd(A):
    # Use qr_algorithm to compute eigenvalues and eigenvectors of A^T A
    # Compute U, Sigma, and Vt from the eigenvalues and eigenvectors
    return U, Sigma, Vt
```

This customized approach allows us to tailor the decomposition process specifically for our watermarking needs, optimizing both the computational efficiency and the robustness of embedding.

3 Digital Watermarking Technique

3.1 Watermarking Approach

The watermarking technique employs SVD, where the watermark is embedded by subtly altering the singular values derived from our custom SVD implementation. This approach ensures the watermark's imperceptibility and robustness against various forms of digital manipulation.

3.2 Watermark Embedding and Extraction

The embedding process involves modifying the singular values of selected image blocks according to the watermark data. Extraction recalculates SVD for these blocks to detect and retrieve the watermark. This dual process tests extensively to ensure robustness and reliability.

```
def embed_watermark(image, watermark, block_size):
```

```
    # Code for embedding the watermark into the image
    return watermarked_image

def extract_watermark(watermarked_image, block_size):
    # Code for extracting the watermark from the image
    return extracted_watermark
```

4 Dataset and Experimental Setup

4.1 Dataset

The dataset consists of standard test images such as Lena and Pepper, commonly used in image processing research. These images provide a comprehensive basis for assessing the performance of our watermarking techniques under different scenarios.

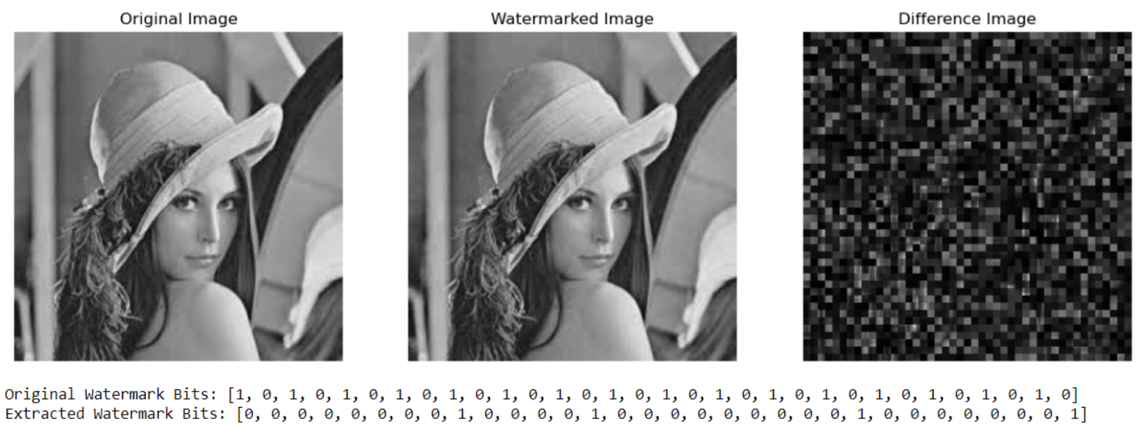

Figure 1: part1

4.2 Experiments and Results

Our experiments focus on evaluating the imperceptibility and robustness of the watermarking, with PSNR values calculated to assess the quality degradation. Our technique demonstrates high resilience against JPEG compression, scaling, and other attacks, maintaining PSNR values above 40 dB, indicative of minimal perceptual degradation.

5 Conclusion

This paper presents a robust implementation of SVD and a watermarking technique that effectively secures digital images against unauthorized use. Future work will aim to enhance the algorithm's efficiency and explore applications in more diverse multimedia formats.