CSE 4128: Image Processing and Computer Vision Laboratory

Image Watermarking and Blending for Grayscale and Color Images

Submitted By

Md Arefin Molla
Roll: 2007050

Submitted To

Dr. Sk. Md. Masudul Ahsan
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Md Tajmilur Rahman
Lecturer
Department of Computer Science and Engineering
Khulna University of Engineering & Technology

Department of Computer Science and Engineering
Khulna University of Engineering & Technology Khulna
9203, Bangladesh
November 5, 2025

Objectives:

1) To create a method for embedding visible and invisible watermarks; not significantly degrading the original image quality.
2) To create seamless image blending process; maintaining integrity and visuals.
3) To build a user-friendly interface allowing easy application of these techniques.

Introduction:

Watermarking and Blending are crucial techniques addressing both security and creative aspects of image handling. Watermarking embeds a mark or logo into an image, serving as a copyright tool without significantly affecting the visual quality. Blending combines multiple images into a single image, maintaining the integrity of visual information from each.

Tools Used:

1) Python – As the base programming language because of its flexibility and useful environment to manipulate images and implement complex image processing algorithms.
2) OpenCV - Used this open-source library for its comprehensive tools set for real-time computer vision, essential for watermarking and blending.
3) Tkinter – Standard GUI toolkit for Python. Used for creating an accessible and interactive interface for users that removed the necessity of writing any code by the user.

Methodology:

Image Watermarking:

Image watermarking involves embedding information into an image, which can be a logo, text, or a pattern, that can identify the creator, owner, or authorized consumer.

Generally, watermarking can be of two types: Visible and Invisible.

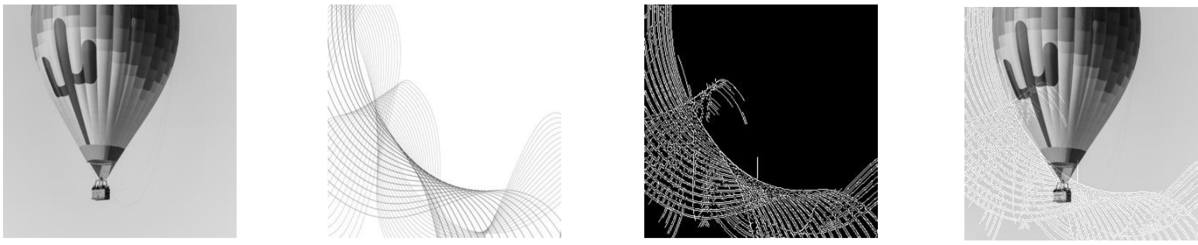
Visible Watermarking:

Places a transparent text or logo over the image, subtle enough not to interfere significantly with the image but clear enough to serve its protective purpose.

Here, for visible watermarking, we have used an edge-detection based approach.

Edge-Detection Based Watermarking:

Firstly, we detect edges in the watermark image using Canny algorithm. Then these edges are overlayed onto the original image in a particular color (gray). We used an ‘edge_opacity’ parameter to control the visibility level.



Resized Main Image

Resized Main Image

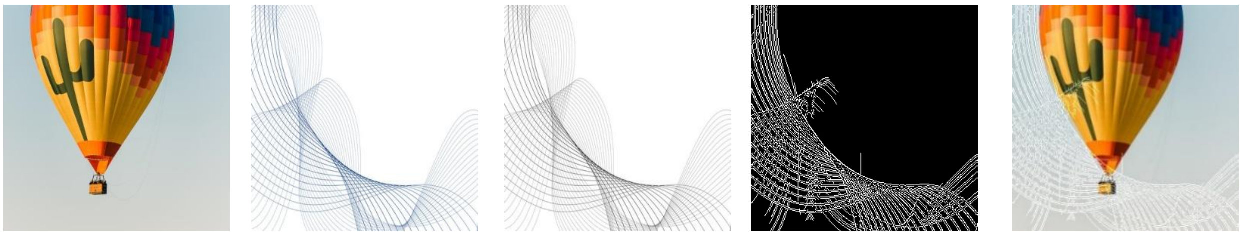
Edge Detected

Watermarked Image

Figure 1: Visible Watermarking using Canny Edge Detection Algorithm (on Grayscale Image)

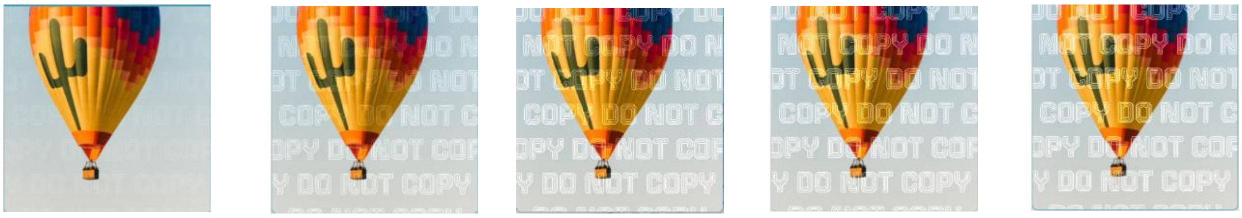
Opacity Parameter (alpha)

By setting the opacity parameter’s value (range 0-100), user can set the visibility level of the watermark.



Resized Main Image | Resized Watermark Image | Grayscaled Watermark Image | Edge Detected | Watermarked Image

Figure 2: Visible Watermarking using Canny Edge Detection Algorithm (on Colour Image)



edge_opacity = 6 | edge_ opacity = 29 | edge_ opacity = 62 | edge_ opacity = 81 | edge_ opacity = 100

Figure 3: Visibility (transparency) of watermark's change with respect to edge_opacity(alpha)

Invisible Watermarking:

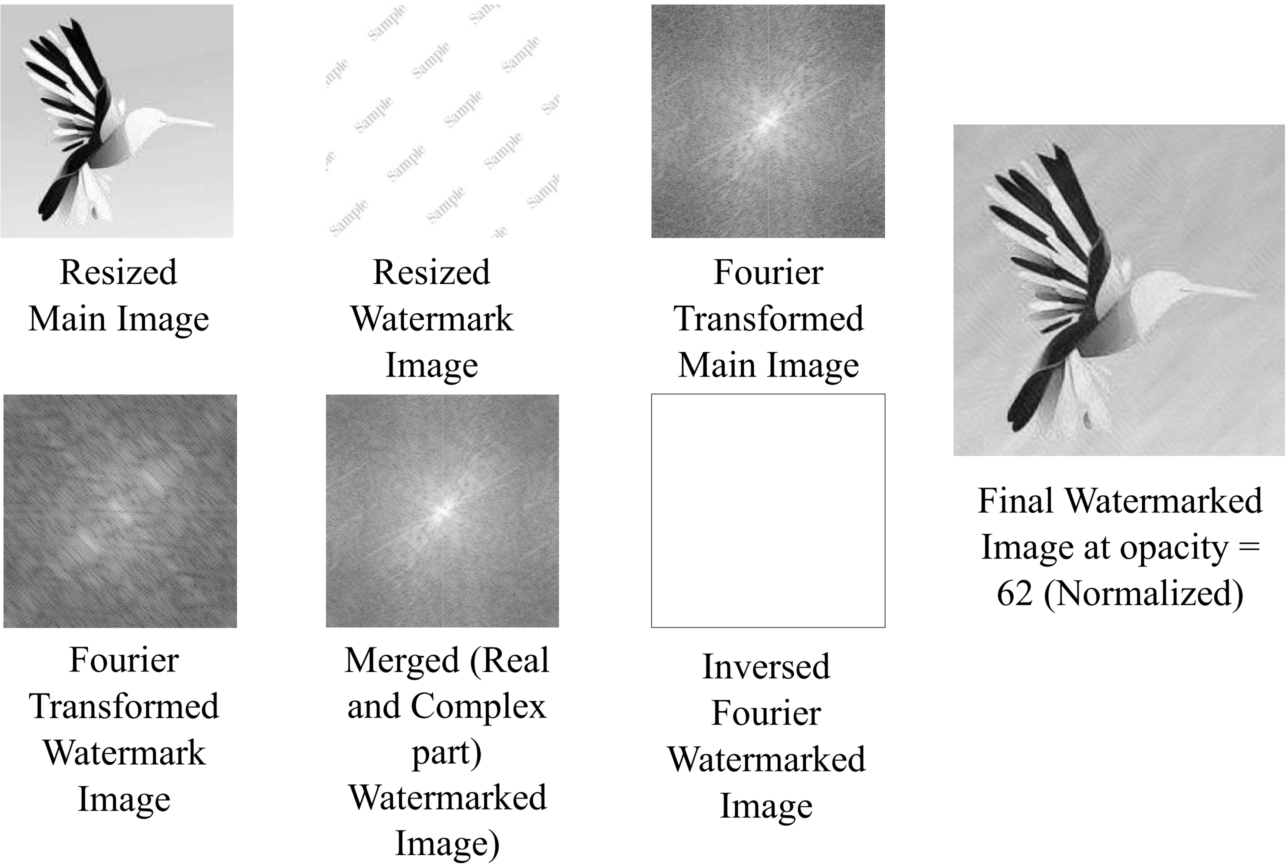
This process uses methods that embed information in the image that is not detectable to the human eye. Possible techniques can be LSB insertion, Fourier Transform etc.

Here, for invisible watermarking, we have used Fourier Transformation based approach.

Fourier Transform based Watermarking:

It embeds watermark in the frequency domain, uses Fourier Transform.

For a color image, it splits into three channels (RGB) and apply the embedding differently on each channel considering each channel as a single Grayscale image, then merge the channels to form the final Watermarked Color Image. Here also, an opacity parameter controls the strength of the watermark embedding.



Resized Main Image

Resized Watermark Image

Fourier Transformed Main Image

Final Watermarked Image at opacity = 62 (Normalized)

Fourier Transformed Watermark Image

Merged (Real and Complex part) Watermarked Image)

Inversed Fourier Watermarked Image

Figure 4: Invisible Watermarking using Fourier Transform (on Grayscale Image)

Watermark Extraction:

Watermark extraction is the process of retrieving a watermark from a watermarked image. The extraction process involves comparing the watermarked image with the original image.

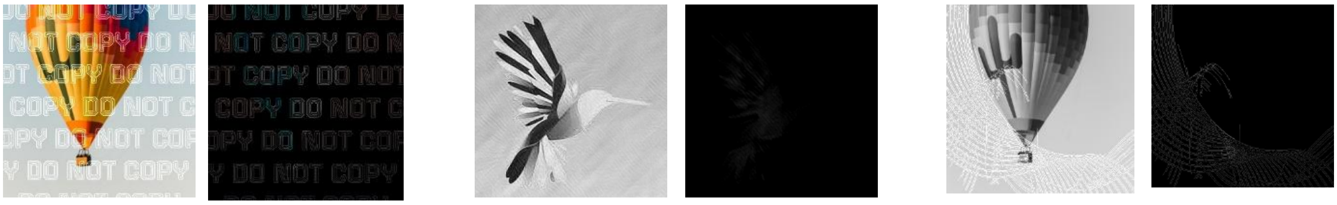
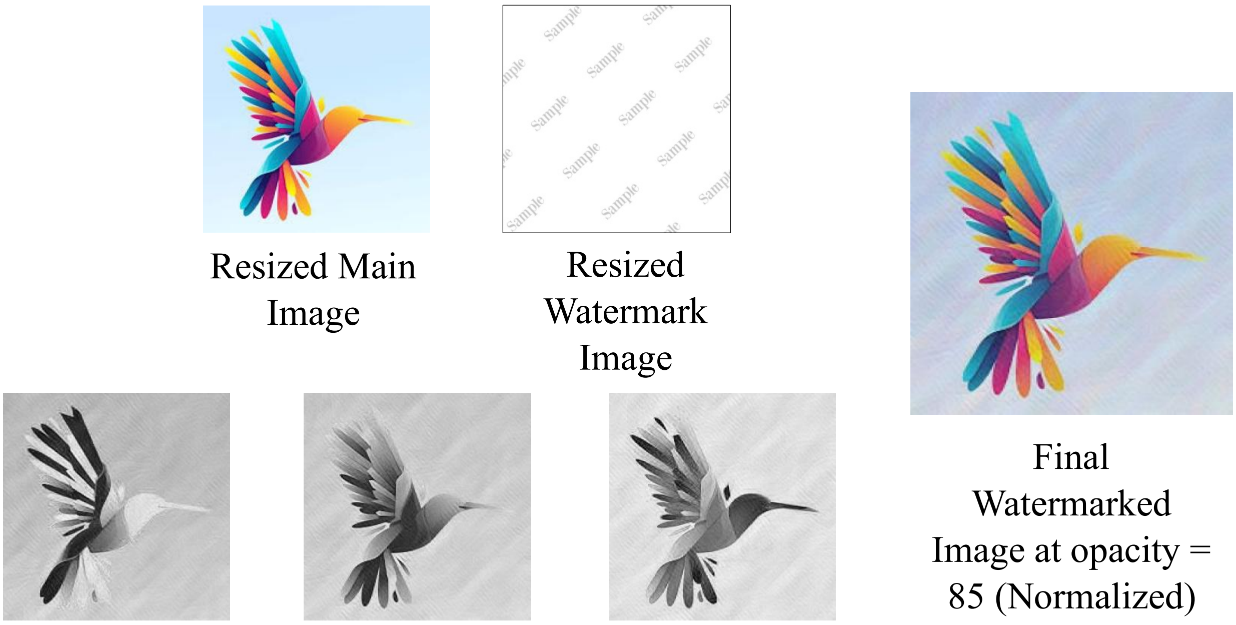


Figure 5: Watermarked Images and their extracted watermark



Resized Main Image

Resized Watermark Image

Final Watermarked Image at opacity = 85 (Normalized)

Final Watermarked (Blue Channel)

Final Watermarked (Green Channel)

Final Watermarked (Red Channel)

Figure 6: Invisible Watermarking using Fourier Transform (on Colour Image)

Image Blending:

Image blending is a technique used in image processing where two or more images are combined together to form a single image.

There are different Blending techniques:

- Gradient Blending
- Directional Blending
- Frequency Domain Blending etc.

Here, Blending is performed using a Gradient mask. This mask controls how one image transitions into the other based on a specified direction and an alpha value (Blending ratio).

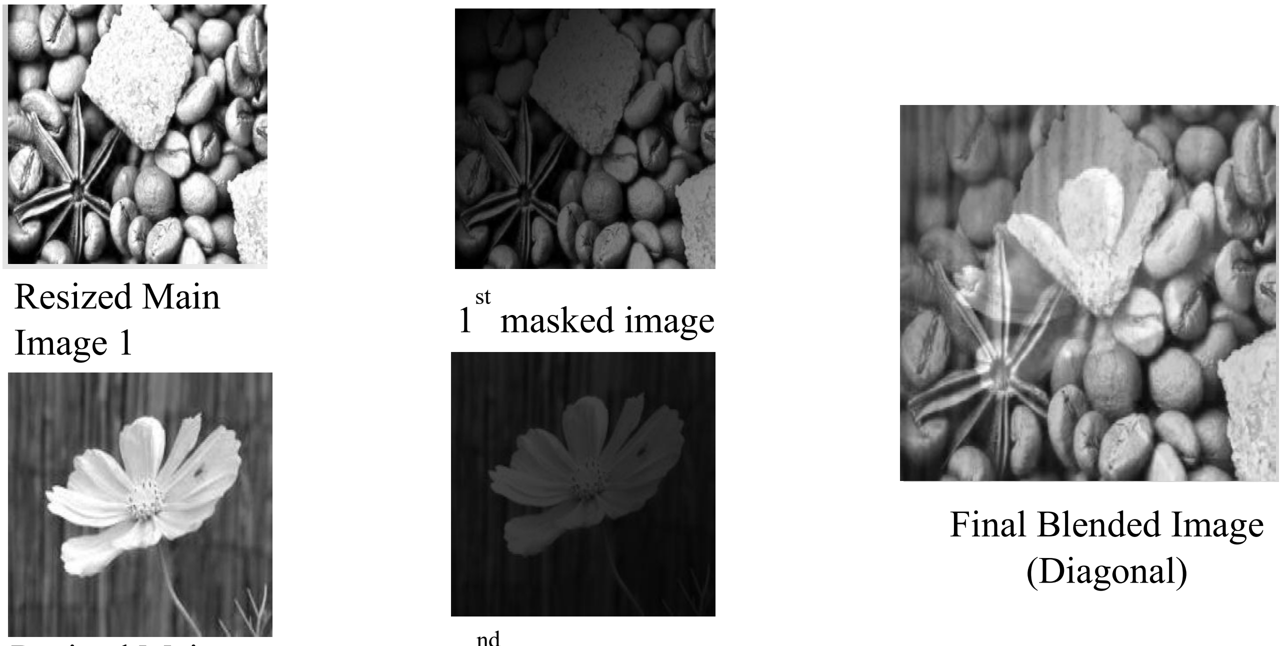
Gradient Mask creation:

- Based on 'direction' parameter, a gradient array is created
- For Left To Right, a horizontal gradient is used where left side of the 1st image is more visible, gradually it shows the more of the 2nd image at right.

- For Top To Bottom, a vertical gradient affects the transition from top (more of 1st image) to bottom (more of 2nd image).
- For Diagonal, a diagonal gradient combines both horizontal and vertical gradients to create a diagonal transition effect.

Mask Application:

- Two masks are derived from the gradient
- 1st mask (mask1) is scaled by alpha, favors the 1st image
- 2nd mask (mask2) is complementary to mask1, favors the 2nd image
- Masks applied to their respective images, the results are summed.
- The sum is element-wise, performed separately for each pixel. (according to gradient and alpha values)



Resized Main Image 1

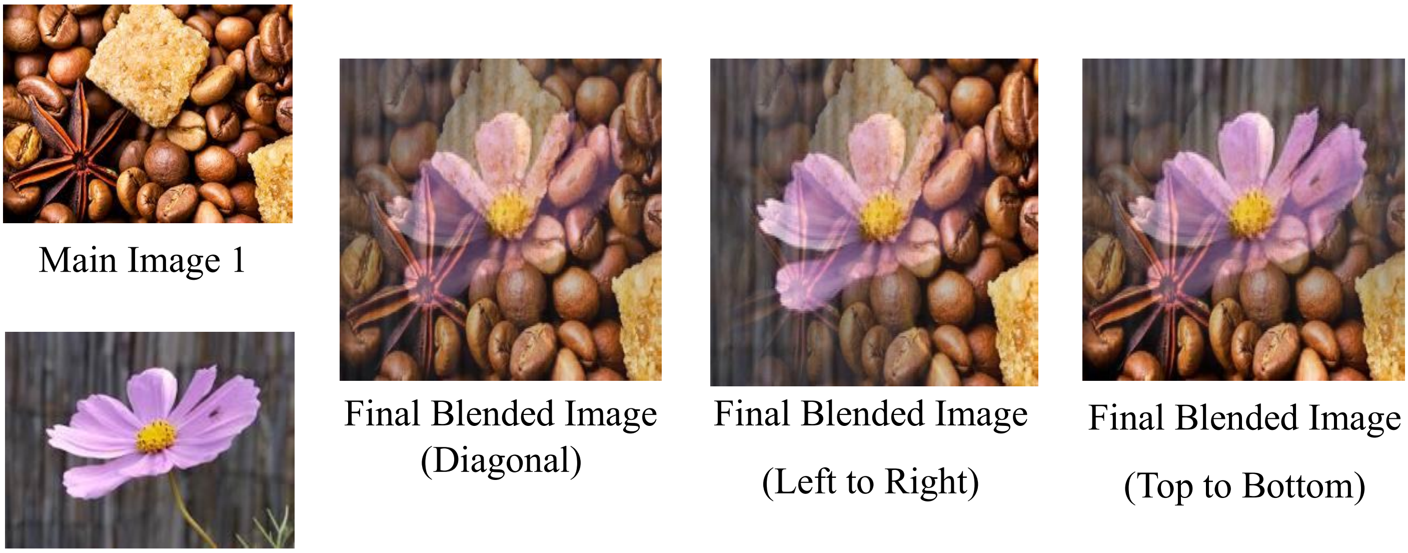
1st masked image

Resized Main Image 2

2nd masked image

Final Blended Image (Diagonal)

Figure 5: Image Blending (Diagonally) on Grayscale Images



Main Image 1

Final Blended Image (Diagonal)

Final Blended Image (Left to Right)

Final Blended Image (Top to Bottom)

Main Image 2

Figure 6: Image Blending (Diagonally, Left to Right, Top to Bottom) on Colour Images

Discussion:

The implementation of watermarking and blending operations using OpenCV and Tkinter has provided a robust and user-friendly interface for manipulating images. The choice of techniques, such as Canny edge detection for visible watermarking and Fourier transform for hidden watermarking, has demonstrated their effectiveness in embedding watermarks. The added capability to handle both grayscale and color images enhances the flexibility of the application, catering to diverse user needs. Additionally, the blending functionality with gradient masks enables smooth transitions between images, making it suitable for various creative and practical applications. The directory structure for saving outputs ensures organized storage, making it easy to retrieve and analyze the results.

Conclusion:

This project successfully combines the power of image processing techniques with a graphical user interface to provide a comprehensive solution for watermarking and blending images. The structured approach to saving outputs in specific directories aids in maintaining clarity and organization, which is crucial for both development and user experience. While the implementation achieves the desired objectives, future enhancements could include more advanced watermarking techniques and additional blending modes to further expand the application's capabilities. Overall, this project serves as a valuable tool for both casual users and professionals in the field of image processing.

References:

1. **Gonzalez, R. C., & Woods, R. E. (2018).** *Digital Image Processing* (4th ed.). Pearson Education. ISBN: 978-0133356724.
2. **Bradski, G., & Kaehler, A. (2008).** *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media. ISBN: 978-0596516130.
3. **Course Materials.** CSE 4128: Image Processing and Computer Vision Laboratory