CENG 466 Introduction to Image Processing Programming Assignment 2

Image Processing Techniques: Edge Detection, image enhancement and image compression

# Cas Carree

*Student ID: 2704930*

***Abstract*—This paper explores three fundamental aspects of digital image processing: pattern extraction using edge detection, image enhancement of noisy images and image compression. For comparison of pattern extraction the Sobel, Roberts and Prewitt edge detectors were utilized in combination with gaussian blurring and most significant bit. Image enhancement was carried out in both the spatial and Fourier domains. In the spatial domain, Gaussian and median filters were applied, while in the Fourier domain, ideal low-pass, band-pass, and band-reject filters were employed. The performance of Haar wavelet and discrete cosine transform were compared for image compression measuring the loss of information using MSE(mean squared error)**

***Keywords*—pattern extraction, edge detection, image enhancement, image compression**

1. Pattern extraction techniques
2. *Background*

Edge detection will be utilized to extract patterns from the images, as these identify points where there is an abrupt change in the image intensity, identifying an edge between two regions [1]. For this, the Sobel, Prewitt, and Roberts operations were used. All these methods are based on calculating the gradient of the image, and based on this can be determined if an edge exists. However these methods are prone to noise to mitigate this Gaussian smoothing and most significant bit were applied. The most significant bit (MSB) corresponds to the highest-value bit in a multi-bit binary number. Typically, it is the leftmost bit or the first bit transmitted in a sequence. For example, in the binary number 1000, the MSB is 1, while in the binary number 0111, the MSB is 0 [2]. By applying edge detection to both blurred and binarized (MSB) images, the interaction between smoothing, binarization, and edge detection can be analyzed. This iterative approach reveals how preprocessing affects the ability of edge detection filters to extract meaningful patterns

1. *Methodology*

The methodology consisted of the following steps:

-**Edge detection on original image:**

Using scikit-image’s filters module Sobel, Roberts and Prewitt filters were applied to the image. Following these operations, the resulting images were saved in the output directory.

**-Blurring the image with Gaussian filters:**  
Using GaussianBlur from OpenCV, the image was smoothed, applying three different kernel sizes (3x3, 5x5, 7x7). The results were saved to be used in the following steps.

**-Edge detection on blurred images:**  
Reapplying Sobel, Roberts, and Prewitt filters to the blurred images, comparing the edge detection results with those from the original image.

**-Extraction of the Most Significant Bit (MSB):**  
Binarized the image by isolating the most significant bit of pixel intensities, creating a high-contrast binary image by thresholding at intensity value 128.

**-Edge detection on MSB images:**  
Applying the same edge detection filters to the MSB image to evaluate the effect of binarization on edge quality.

**-Blurring the MSB images:**  
Reapplying Gaussian blurring to the MSB images with the same kernel sizes, generating additional smoothed outputs.

-E**dge detection on blurred MSB images:**  
Performed edge detection on each blurred MSB image to examine the combined impact of blurring and binarization on edge detection.

1. *Results and Limitations*

The analysis of the results by visual inspection revealed that all edge detection filters effectively captured the main patterns in the images. However, smoothing significantly reduced the detection of finer patterns, with this effect being most pronounced in the MSB (Most Significant Bit) case. The Gaussian smoothing, in particular, introduced an additional downside: edges appeared thicker than they actually were, potentially altering the perception of the patterns. Due to the relatively low presence of noise in the images, applying the edge detection filters without any smoothing was sufficient to extract the patterns clearly. This suggests that smoothing may not always be necessary in scenarios with minimal noise, and its use should be carefully evaluated to avoid unnecessary degradation of fine details.

1. image enhancement
2. *Background*

Noise is an inherent issue in image processing. To tackle this problem various filtering techniques are employed to reduce noise and enhance image quality. These techniques include spatial filters, such as median and Gaussian filters, and frequency-based methods, such as low-pass, band-pass, and band-reject filters. Each approach is designed to target specific noise characteristics while preserving essential image features. For example the median filter is used for handling salt and pepper noise, which are randomly occurring bright and dark pixels which can severely impact the image quality [3]. This is due to its robustness against outliers as it only takes the median value of the kernel into account [3]. To remove subtle random noise the Gaussian filter is used. This filter reduces noise by applying a Gaussian kernel to the image which is based on the gaussian distribution and gives more weight to the central pixel to retain the image’s structure [4].One of its limitations is it incorporates noise into the result, causing indiscriminate smoothing along edges [4]. To circumvent this filtering in the frequency domain might be beneficial. This can be done by a Low-pass filter which operates by allowing low-frequency components to pass while suppressing high-frequency details, such as noise and fine textures. This approach preserves the overall structure of an image while smoothing sharp variations, enhancing visual quality [5]. The band-pass filter works similarly to the low-pass filter however this one has an upper and lower bound so only frequencies between a certain range are passed through. This selective filtering emphasizes mid-frequency details, making it ideal for highlighting textures and patterns while reducing low and high-frequency noise. The band-reject filter does the opposite of the bandpass filter, by only rejecting the frequencies within a specific range. This can be beneficial to eliminate noise appearing in only a specific frequency band. In conclusion, the choice of filtering technique depends on the specific noise characteristics and the desired balance between noise reduction and feature preservation. Each method brings unique strengths and limitations, underscoring the importance of selecting the appropriate filter for the application at hand.

1. *Methodology*

Our implementation involved several critical steps:

Noise Analysis in Spatial Domain:

* + Using cv2’s split function the images were split into RGB channels for noise analysis. Plotting the grayscale images of each channel to assess the noise within each channel.

Noise Analysis in Fourier Domain:

* + Using Scipy’s discrete Fourier transform and shift function the Fourier Transform for each channel was calculated. Each of these was plotted again to analyze the type of noise

Median Filtering:

* + Applying cv2’s median blur function to each color channel using kernel sizes of 3, 5, 7, and 9 pixels.
  + Merging the filtered channels bank into full-color images and saving the results.

Gaussian Filtering:

* + Applying cv2’s Gaussian blur to each channel with varying kernel sizes (3x3, 5x5, 7x7, 9x9).
  + Recombined the blurred channels into full-color images to observe the smoothing effects.

Ideal Low-Pass Filtering:

* + Designed a circular low-pass filter to retain low-frequency components below a specific radius in the Fourier domain. This was done by calculating the Euclidean distance for each pixel and keeping only the ones below the threshold. This low-pass filter was applied to each channel and afterwards those were merged back into the original image

Band-Pass Filtering:

• Designed a circular band-pass filter to isolate frequency components within a specific range (low-radius to high-radius) in the Fourier domain. This was achieved by calculating the Euclidean distance for each pixel and retaining only those within the specified frequency range. The band-pass filter was applied separately to each channel, and the filtered channels were then combined to reconstruct the image, highlighting mid-frequency features.

Band-Reject Filtering:

• Designed a circular band-reject filter to attenuate frequencies within a specific range in the Fourier domain. This was achieved by defining an inner and outer radius and calculating the Euclidean distance for each pixel. Frequencies falling within this range were suppressed while others were retained. The filter was applied independently to each channel, effectively reducing noise while preserving essential image details, and the filtered channels were then merged to reconstruct the image.

1. *Results and Limitations*

Due to lack of expertise in the analysis of noise in the Fourier domain the design of the algorithm and selection of parameters was limited. This was translated into the limited success of image enhancement. The first image, likely affected by salt-and-pepper noise, was effectively cleaned using the median filter. However, satisfactory enhancement was not achieved for the second and third images. Several factors contributed to this outcome. Firstly, the same filter and kernel were applied to each channel, despite the noise characteristics differing between channels, reducing the filters' effectiveness. This issue was particularly pronounced in the third image, where each channel exhibited distinct periodic noise patterns. Secondly, the band-pass and band-reject filters were not optimized for the specific noise in the images. For instance, the green channel of the third image contained prominent horizontal noise in the frequency domain, but the circular design of the filters' masks failed to address this effectively, limiting their ability to remove such noise. In conclusion, achieving effective image enhancement requires tailoring filters and parameters to the specific noise characteristics of each channel, highlighting the importance of noise analysis expertise and adaptive algorithm design for optimal results.

1. Image compression
2. *Background*

Image compression is a critical aspect of modern digital imaging, aiming to reduce file sizes while preserving essential image details [6]. Transform-based methods, such as the Discrete Cosine Transform (DCT) and Wavelet Transform, are commonly employed due to their efficiency and flexibility in representing image data in a compact form. The **Wavelet Transform** provides a multi-resolution analysis of an image by decomposing it into sub-bands representing different frequency components. This technique uses wavelets, such as the Haar wavelet, to separate an image into Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH) sub-bands [7]. The LL sub-band captures the most significant low-frequency content, while the other sub-bands represent higher-frequency details. The **Discrete Cosine Transform (DCT),** on the other hand, converts an image from the spatial domain to the frequency domain by representing pixel intensities as a sum of cosine functions at different frequencies [8]. DCT concentrates the majority of energy into a few low-frequency coefficients, making it particularly effective for compression [8]. In both approaches, **retaining the most significant coefficients** is key to reducing data size while minimizing visual loss. Sorting coefficients by magnitude and retaining the largest ones allows the representation of essential image features while discarding less critical information. Reconstruction involves reversing the transformations—reassembling coefficients into their original or reduced forms and applying the inverse transformation to recreate the spatial-domain image.

1. *Methodology*

Transforming the image for compression:

• For the wavelet transform, the pywt.dwt2 function was used with the Haar wavelet to decompose the image into four sub-bands: Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH).

•For the DCT (Discrete Cosine Transform), a 2D DCT was applied using scipy.fftpack.dct with orthonormal normalization to convert the image into the frequency domain.

Retaining the most significant coefficients:

• The coefficients from the selected transform were

flattened into a single array.

• Based on the desired percentage, the top k coefficients with the largest magnitudes were identified using np.argsort.

• A new coefficient array was created, retaining only the largest k coefficients and setting the rest to zero.

Reconstruction of compressed coefficients:

• For the wavelet transform, the retained coefficients were reshaped into the original LL, LH, HL, and HH sub-band shapes, forming the compressed wavelet representation.

•For the DCT, the retained coefficients were reshaped to match the original image's shape.

Reconstructing the image:

• For the wavelet transform, the pywt.idwt2 function was used with the compressed wavelet coefficients to reconstruct the image.

•For the DCT, the inverse DCT was applied using scipy.fftpack.idct with orthonormal normalization to revert the coefficients back to the spatial domain.

1. *Results and limitations*

The Haar Wavelet Transform achieves significantly better compression ratios compared to the Discrete Cosine Transform (DCT). For Image 1, Haar produces file sizes of 12.4 KB, 23.4 KB, and 189 KB for N=1%,10%, and 50%, respectively, while DCT results in larger sizes of 25.9 KB, 204 KB, and 762 KB. For Image 2, Haar compresses to 21.9 KB, 78 KB, and 585 KB, while DCT results in sizes of 161 KB, 1.08 MB, and 3.83 MB for the same N values. Similarly, for Image 3, Haar achieves file sizes of 5.82 KB, 11.7 KB, and 177 KB compared to DCT’s 48.1 KB, 322 KB, and 1.08 MB. This demonstrates Haar’s superiority in reducing file sizes. However, compression alone is insufficient if it results in excessive loss of image information; thus, the MSE of the two algorithms was compared to assess the quality of the reconstructed images.

Loss of Information (MSE and Visual Quality):

• At N=1%: Haar exhibits higher MSE (19662 for Image 1) compared to DCT (324), resulting in noticeable artifacts making the image unrecognizable. DCT produces visually smoother reconstructions.

• At N=10%: Haar’s MSE reduces (5816 for Image 1) but remains higher than DCT’s (77), reflecting better fidelity in DCT outputs.

• At N=50%: Both methods achieve high-quality results, with Haar often slightly outperforming DCT in MSE (1.57 vs. 4.73 for Image 1)

This pattern is repeated in the other 2 images. To summarize Haar Wavelet excels in compression efficiency, making it ideal for applications prioritizing file size. DCT, with its superior fidelity, is better suited for scenarios requiring high visual quality at moderate compression ratios. The choice depends on the desired trade-off between compression and image clarity.

1. Conclusion

This report highlights the trade-offs and challenges in three areas of digital image processing: pattern extraction, image enhancement, and image compression. For pattern extraction, edge detection filters effectively captured the main features, but the introduction of Gaussian smoothing and binarization revealed limitations in preserving finer details, particularly in low-noise scenarios. In image enhancement, the success of noise reduction techniques varied significantly based on noise characteristics, emphasizing the need for precise noise analysis and adaptive filter design. Finally, the comparison between Haar Wavelet and DCT for image compression demonstrated Haar’s superiority in achieving high compression ratios, while DCT provided better visual fidelity in reconstructed images. These findings underline the importance of tailoring techniques to specific application requirements, balancing performance, and preserving critical image details to optimize outcomes in digital image processing tasks.

# References:

|  |  |
| --- | --- |
| [1] | geeksforgeeks, „geeksforgeeks.org,” 10 6 2024. [Online]. Available: Feature Extraction in Image Processing: Techniques and Applications - GeeksforGeeks. |
| [2] | R. Sheldon, „techtarget.com,” [Online]. Available: https://www.techtarget.com/whatis/definition/most-significant-bit-or-byte. [Geopend 13 12 2024]. |
| [3] | S. Gonzalez, „blog.salvatorelabs.com,” 5 8 2023. [Online]. Available: https://blog.salvatorelabs.com/understanding-the-median-filter-a-powerful-tool-for-noise-reduction-in-image-processing/?utm\_source=chatgpt.com. |
| [4] | H. Vedoveli, „medium.com,” 26 8 2023. [Online]. Available: https://medium.com/@henriquevedoveli/image-filtering-techniques-in-image-processing-part-1-d03362fc73b7. |
| [5] | B. Williams, 7 8 2023. [Online]. Available: https://gadgetsranked.com/what-is-ideal-low-pass-filter-in-image-processing/. |
| [6] | A. V. D. I. M. A. A.-k. Murooj Khalid Ibraheem Ibraheem, „iieta.org,” 16 5 2024. [Online]. Available: https://iieta.org/journals/isi/paper/10.18280/isi.290307. |
| [7] | uotechnlogy, „uotechnology.edu,” [Online]. Available: https://uotechnology.edu.iq/ce/lecture%202013n/4th%20Image%20Processing%20\_Lectures/DIP\_Lecture9.pdf. [Geopend 13 12 2024]. |
| [8] | geeksforgeeks, „geeksforgeeks.com,” 5 8 2022. [Online]. Available: https://www.geeksforgeeks.org/discrete-cosine-transform-algorithm-program/. |