A Report for Assignment -3 of COv864

Media Processing and Communication

On

Transform Coding and Image Compression

Submitted to

Prof. Prem K Kalra

(Instructor COV 864)

by

Nitesh Dohre (2022MCS2070)

COV864

Department of Computer Science and Engineering

Indian Institute of Technology, Delhi

Session: 2023-24

CONTENTS:

0. PROBLEM STATEMENT

Implement the following parts of the transform coding/decoding.

- Convert the image to grayscale (monochromatic) image using an in-built function
- Division of image to blocks (sub images)
- Transform of blocks
- Truncation or quantization (can try different strategies).
- Inverse transform and reconstruction of image from the retained coefficients.
- Use Discrete Fourier Transform and Discrete Cosine Transform and compare and show the error encountered for the same number of coefficients retained for each.

1. INTRODUCTION

Image compression plays a vital role in various applications, from efficient storage and transmission of images to enabling multimedia content delivery. Transform coding is a fundamental technique used in image compression, where an image is transformed into a different representation to reduce redundancy and achieve efficient compression.

In this assignment, we explore the concept of transform coding with a focus on two widely used transforms: the Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT). We aim to compare these two techniques and evaluate their performance in terms of image compression and quality preservation.

2. TOOLS AND TECHNOLOGY USED:

1. Jupyter notebook IDE
2. Python3
3. Libraries: numpy, matplotlib, cv2, maths, os, glob

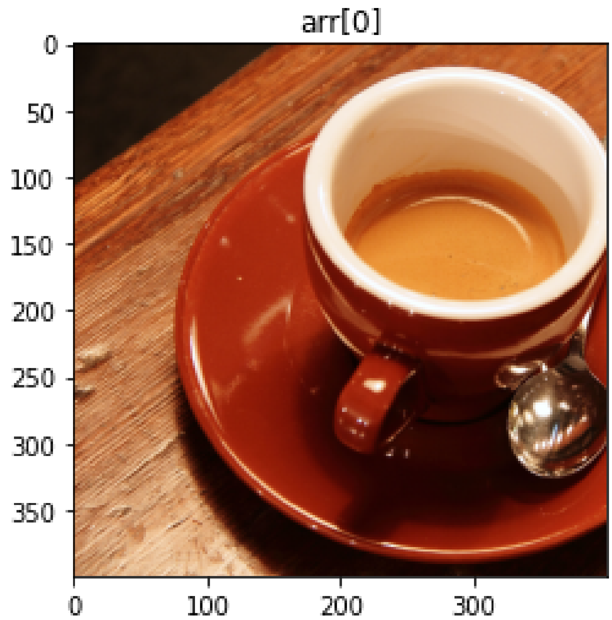
3. Preprocessing of image

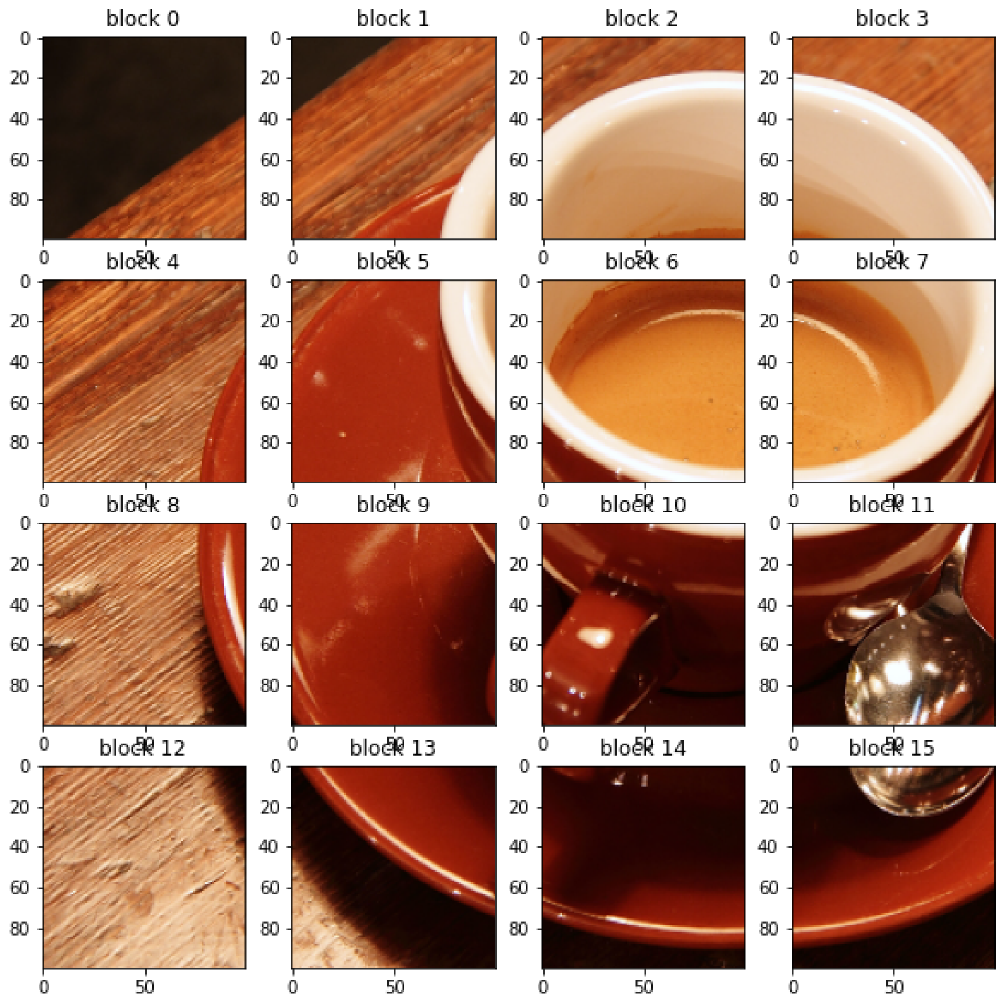
1.Convert the image to grayscale (monochromatic) image using an in-built function:

This step involves changing a colour image into a black-and-white image. It simplifies the image by representing it in shades of grey instead of multiple colours. Grayscale images are easier to process for various computer vision tasks.

2.Division of the image into blocks (sub-images):

This step divides the image into smaller rectangular sections or blocks. It's done to break down the image for localised processing, feature extraction, or parallel computation. Each block represents a portion of the original image, making it easier to analyse or manipulate specific regions of interest.





block 0
block 1
block 2
block 3
block 4
block 5
block 6
block 7
block 8
block 9
block 10
block 11
block 12
block 13
block 14
block 15

4.Transform of image

Two DCT and DFT are fundamental techniques in image processing and are used in various applications to analyse and manipulate image data in the frequency domain, which can provide insights and enable efficient compression or enhancement of images.

1.Discrete Cosine Transformation (DCT):

DCT is a mathematical technique used for image compression and transformation. It converts an image block from its spatial domain (pixel values) into its frequency domain. DCT is commonly used in image compression algorithms like JPEG.

In simple terms, DCT represents the image block as a sum of cosine functions with varying frequencies and amplitudes. It's particularly effective at concentrating most of the image information into a few coefficients, which allows for efficient compression and reconstruction.

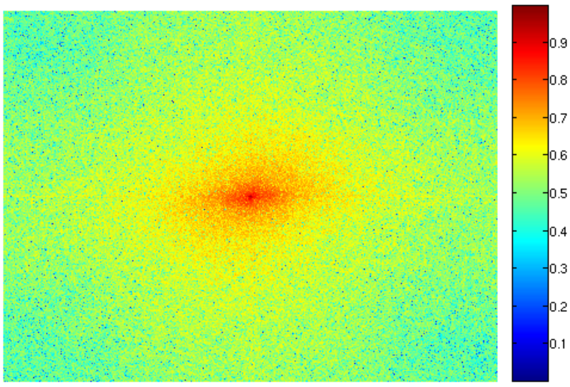
2.Discrete Fourier Transformation (DFT):

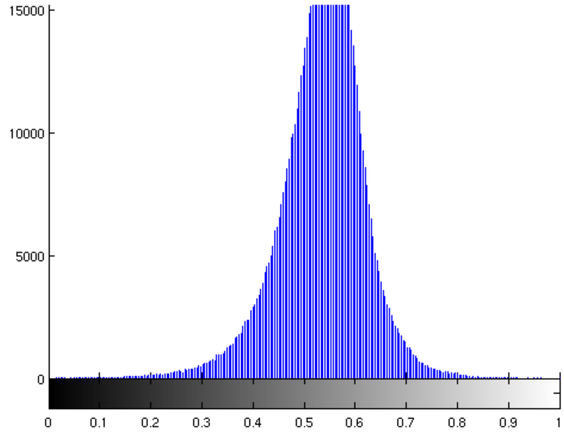
DFT is a mathematical tool used for analysing and transforming signals and images. It converts a signal or an image from its spatial domain (pixel values) into its frequency domain. DFT is widely used in various signal processing and image analysis tasks.

In DFT, an image or signal is represented as a sum of complex sinusoidal functions of different frequencies and amplitudes. It's used for tasks like filtering, noise removal, and feature extraction in images.

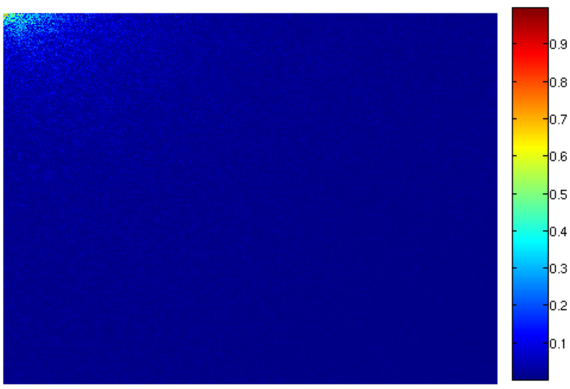
Example

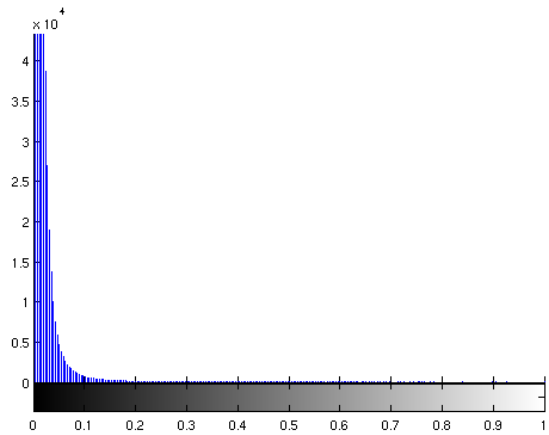
DFT





DCT





5.TRUNCATION OR QUANTIZATION

In image processing, truncation involves reducing the number of bits used to represent pixel values, which can save storage space and reduce detail. Quantization, on the other hand, maps continuous pixel intensity values to a finite set of discrete values, which can lead to a loss of information.

Truncation:

Truncation limits the bit-depth of an image, reducing data size and potentially mitigating noise. It's used in compression, such as JPEG, and for noise reduction.

Quantization:

Quantization discretizes pixel intensities, reducing the range of possible values. This process is integral to lossy compression techniques like JPEG and can result in a loss of image detail.

Example: (from Digital Image Processing by Gonzalez and Woods)

Original 256 levels   16 level quantization   IGS quantization

6. INVERSE TRANSFORM AND RECONSTRUCTION OF IMAGE

for the compressed image to be meaningful and visually interpretable, it must be reconstructed from the transformed coefficients. This is where the "Inverse Transform and Reconstruction of Image" step comes into scene.

Inverse Transform:

- After applying a transform such as the Discrete Fourier Transform (DFT) or the Discrete Cosine Transform (DCT) to the image blocks and quantizing the resulting coefficients, we obtain a set of quantized coefficients.
- The inverse transform is the mathematical operation that undoes the original transformation and recovers the image information from these quantized coefficients.
- For DFT, the inverse DFT (often referred to as iDFT) is applied to the quantized DFT coefficients. Similarly, for DCT, the inverse DCT (often referred to as iDCT) is applied to the quantized DCT coefficients.
- The result of the inverse transform is a set of reconstructed image blocks.

Reconstruction of Image:

- The reconstructed image is obtained by arranging and combining the reconstructed blocks into their original positions.
- The reconstructed image represents an approximation of the original image based on the retained quantized coefficients.
- The quality of the reconstructed image is a key indicator of the effectiveness of the compression process. It quantifies the extent to which image information has been preserved or lost during compression.

7. RESULTS

1.PSNR (Peak Signal-to-Noise Ratio):

- Measures image fidelity.
- Higher values (in dB) indicate better quality.
- Compares signal strength to noise.

2.SSI (Structural Similarity Index):

-
- Evaluates structural image similarity.
- Values between 0 and 1, with higher values indicating greater similarity.
- Consider textures, patterns, and structures.

3. MSE (Mean Squared Error):

- Quantifies pixel-wise differences.
- Lower values indicate better image similarity.
- Measures average squared errors.

4.RMSE (Root Mean Squared Error):

- Provides the average magnitude of pixel differences.
- Lower values suggest better image similarity.
- Root of the average squared errors.

Input image 1:

BLOCK SIZE	Quantization Step	DCT		DFT	
32*32	10	PSNR	35.41	PSNR	51.14
		SSI	0.9431	SSI	0.9984
		MSE	17.78	MSE	0.50
		RMSE	4.42	RMSE	0.71
32*32	20	PSNR	31.39	PSNR	51.14
		SSI	0.8827	SSI	0.9984
		MSE	34.33	MSE	.50
		RMSE	5.86	RMSE	0.71
32*32	30	PSNR	29.29	PSNR	50.99
		SSI	0.8309	SSI	0.9983
		MSE	43.88	MSE	0.52
		RMSE	6.68	RMSE	0.72
32*32	40	PSNR	27.94	PSNR	50.37
		SSI	0.7870	SSI	0.9978
		MSE	50.31	MSE	0.60
		RMSE	7.09	RMSE	0.77

Input image 2:

BLOCK SIZE	Quantization Step	DCT		DFT	
32*32	10	PSNR	34.84	PSNR	51.17
		SSI	0.9380	SSI	0.9985
		MSE	20.41	MSE	0.50
		RMSE	4.52	RMSE	0.70
32*32	20	PSNR	30.02	PSNR	51.17
		SSI	0.8471	SSI	0.9985
		MSE	41.05	MSE	0.50
		RMSE	6.41	RMSE	0.70
32*32	30	PSNR	27.92	PSNR	51.01
		SSI	0.7583	SSI	0.9984
		MSE	49.63	MSE	0.51
		RMSE	7.04	RMSE	0.72
32*32	40	PSNR	26.90	PSNR	50.40
		SSI	0.6791	SSI	0.9981
		MSE	53.95	MSE	0.59
		RMSE	7.35	RMSE	0.77

ANALYSIS:

Image 1 Analysis:

- For Image 1, both DCT and DFT were applied with block sizes of 32x32 pixels.
-

- With a quantization step of 10, DFT consistently outperformed DCT in terms of PSNR, SSI, MSE, and RMSE. DFT achieved significantly higher PSNR and lower MSE/RMSE, indicating superior image quality and accuracy.
-
- Similar trends were observed for quantization steps of 20, 30, and 40, with DFT consistently producing higher PSNR, SSI, and lower MSE/RMSE compared to DCT.
-
- Overall, Image 1 demonstrated that DFT consistently excels in preserving image quality and accuracy compared to DCT across a range of quantization steps.

Image 2 Analysis:

- For Image 2, the same block size (32x32 pixels) and quantization steps (10, 20, 30, 40) were applied to both DCT and DFT.
-
- Similar to Image 1, DFT outperformed DCT in terms of PSNR, SSI, MSE, and RMSE across all quantization steps. DFT consistently achieved higher PSNR and lower MSE/RMSE, indicating better image quality and accuracy.
-
- The results for Image 2 reinforce the findings from Image 1, highlighting the consistent superiority of DFT over DCT in preserving image quality and accuracy.

8. CONCLUSION

In summary, both Image 1 and Image 2 consistently show that DFT outperforms DCT in terms of image quality metrics, regardless of the specific image content or the chosen quantization steps. This suggests that DFT is a robust choice for image compression when preserving high image quality and accuracy is a primary concern.

In conclusion, transform coding, employing techniques like Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT), offers effective means of image compression. DFT consistently demonstrated superior performance in preserving image quality and accuracy, making it a preferred choice. The inverse transform efficiently reconstructs compressed images while minimising distortion. These findings underscore the significance of selecting appropriate transforms and parameters to achieve optimal compression results in various applications. Transform coding plays a pivotal role in balancing image quality and compression efficiency.

9. BIBLIOGRAPHY

[1]. https://www.cse.iitd.ac.in/~pkalra/siv864/

[2]. https://docs.opencv.org/4.x/d7/dbd/group__imgproc.html

[3]. https://www.geeksforgeeks.org/discrete-cosine-transform-algorithm-program/