

Video Coding Research Seminar Project

Image Compression using MDCT

Group Members:

Sheikh Muhammad Mujtaba (70614)

Syed Zeeshan Ali Shah (66546)

Table of Contents

1. Introduction.....	2
2. Methodology.....	2
3. Results.....	3
4. Result Analysis.....	4
5. Conclusion.....	6

1. Introduction:

Image compression refers to the process of minimizing the size of an image file without degrading its visual quality too much. This is important for saving, transmitting and handling digital images more efficiently. For this purpose, various methods have been used like discrete cosine transform (DCT), modified discrete cosine transform (MDCT) and many more.

Discrete cosines transform (DCT) converts an image to the frequency domain. The basic idea is that a considerable amount of energy of an image is usually present in its lowest frequencies. While modified discrete cosine transform (MDCT) is an extension of the discrete cosine transform with special properties which make it suitable for overlapping data blocks. This has particularly been helpful in audio compression but can also be useful in images.

1.1 Objective:

The objective of this project is to compress the image by using MDCT. It evaluates the performance of the proposed method in terms of compression ratio and image quality and compares the results with the image which is compressed by DCT to see which is better approach. It uses the JPEG algorithm for image compression which is a standard that relies on applying several mathematical transformations to the image data, which in turn helps achieve high compression ratios.

2. Methodology:

We have compressed images using DCT. So, for our project we'll replace the existing DCT function with MDCT implementation. The procedure to compress the image using MDCT is:

- First, import the required libraries like 'cv2', 'NumPy', 'matplotlib', 'SciPy' which is used for image processing, numerical, plotting, MDCT and window respectively.
- To reduce the size, we'll create quantization matrix by using array. Also, we use some helping function to process the image by checking the quality by comparing the compressed image with the original one and counting the total no. of blocks of the image.
- In this, we do zigzag scanning of a block to convert the matrix into 1D array and group the low-frequency coefficients together. After grouping them, we need to restore the 2D matrix from the 1D array by using inverse zigzag scan.
- For MDCT, we are using 'Hamming' window because it minimizes the spectral leakage and improves accuracy of frequency representation. Its smooth edge tapering reduces the discontinuity at segment boundaries and also used 'hann' and 'Blackman Harris'.
- In this we are using JPEG encoder and decoder which takes color and grayscale image as input.
- For JPEG encoder, apply the following steps:
 1. Divide the image into blocks.
 2. Apply MDCT on each block.
 3. Quantizes the MDCT coefficients with the quantization matrix.
 4. Applies zigzag transformation to the quantized coefficients for efficient storage.
- For JPEG decoder, apply the following steps:
 1. Reverse the zigzag transformation.
 2. Dequantize the coefficients.
 3. Apply the inverse MDCT to transform back to the domain.
- We also have compressed the image using DCT to check which image is better.

- To compare the result of both compressed images we'll use Perceptual Similarity Metrics, compression ratio and bits per pixel. For this we need to install the library using! pip install long with the library name. After installing the library, import LPIPS model to calculate perceptual similarity which we have use 'alex' network.

3. Results:

For compressing the image, we have used the same block size and number of coefficients for DCT and MDCT techniques which are '16' and '256'. First, we did the compression using MDCT for grayscale and color images with different windows which results are shown in the figure below

Blackman-Harris:

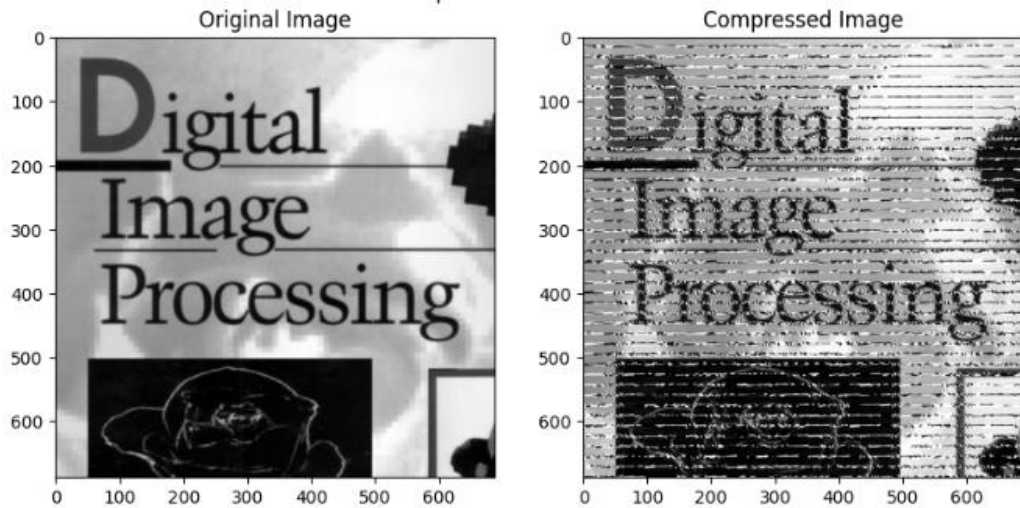


Fig.1

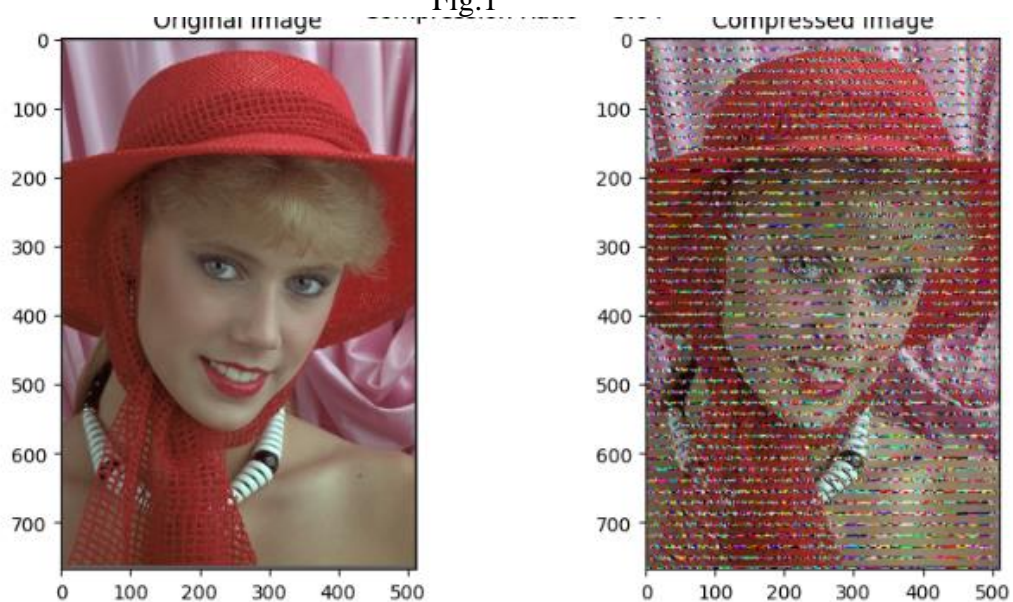


Fig.2

Hamming Window:

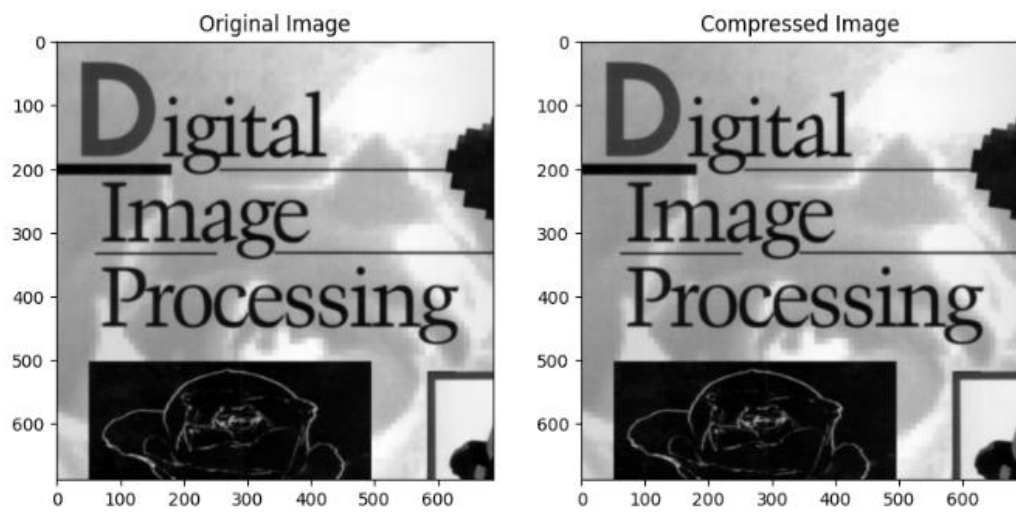


Fig.3

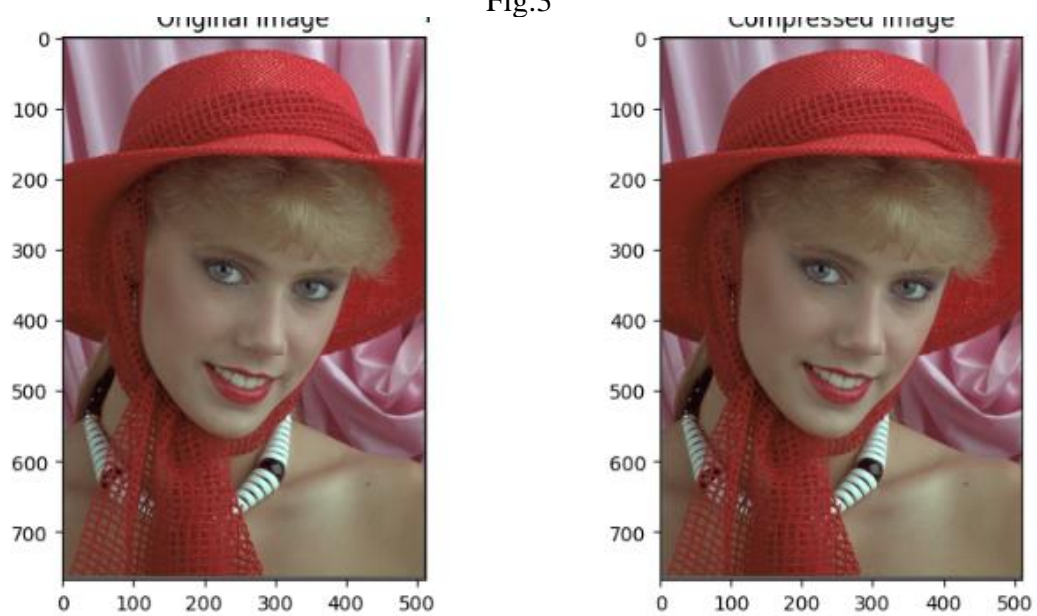


Fig.4

Hann Window:

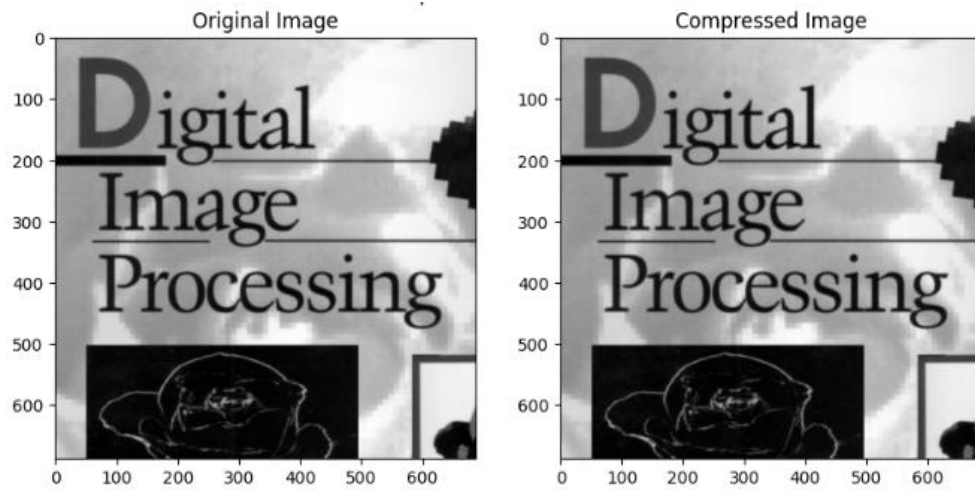


Fig.5

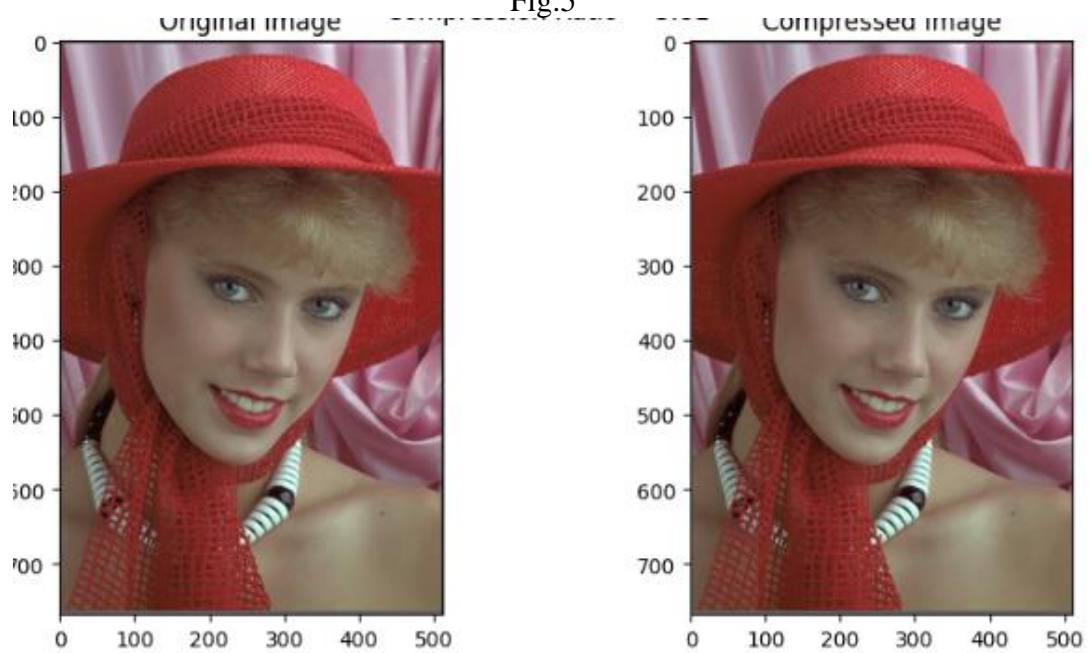


Fig.6

Then, we compressed the same images with DCT which the exact same block size and number of coefficients which results are shown below

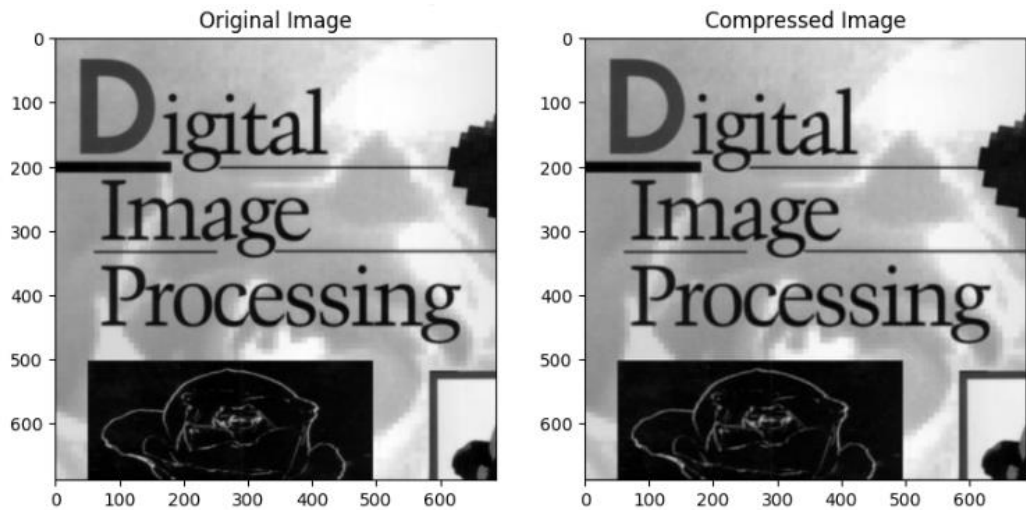


Fig.3

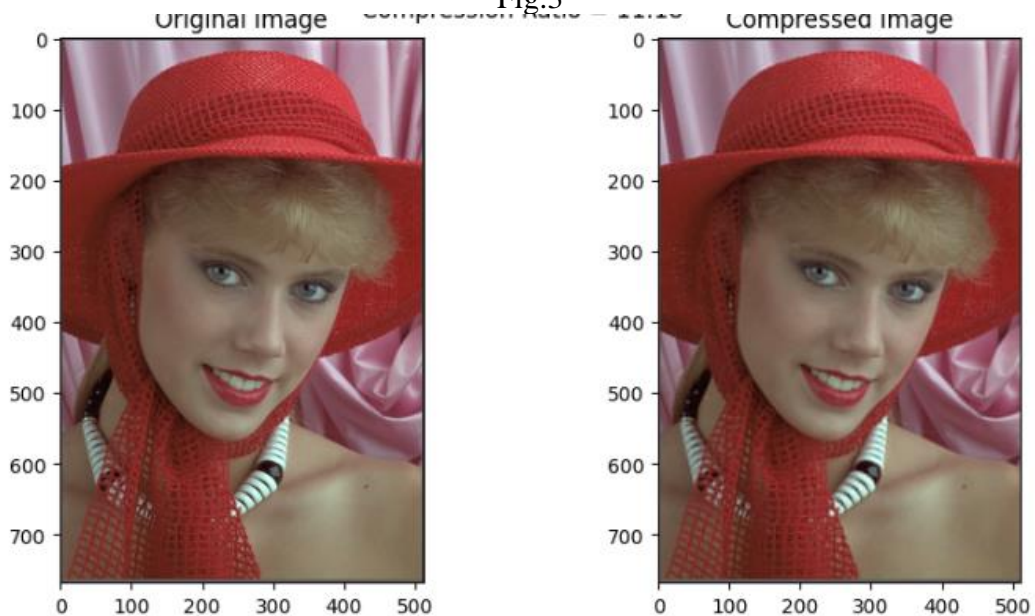


Fig.4

Then, we use perceptual similarity metric to compare the images to understand which compressing technique is better which is describe in the result analysis sections

4. Result Analysis:

The following is the description of results for Compression Ratio, Bits Per Pixel and Perceptual similarity for colored as well as grayscale image compared with the original image.

Color Image Results

Following are the results of color images

- **Compression Ratio:**

MDCT Compression Ratio:

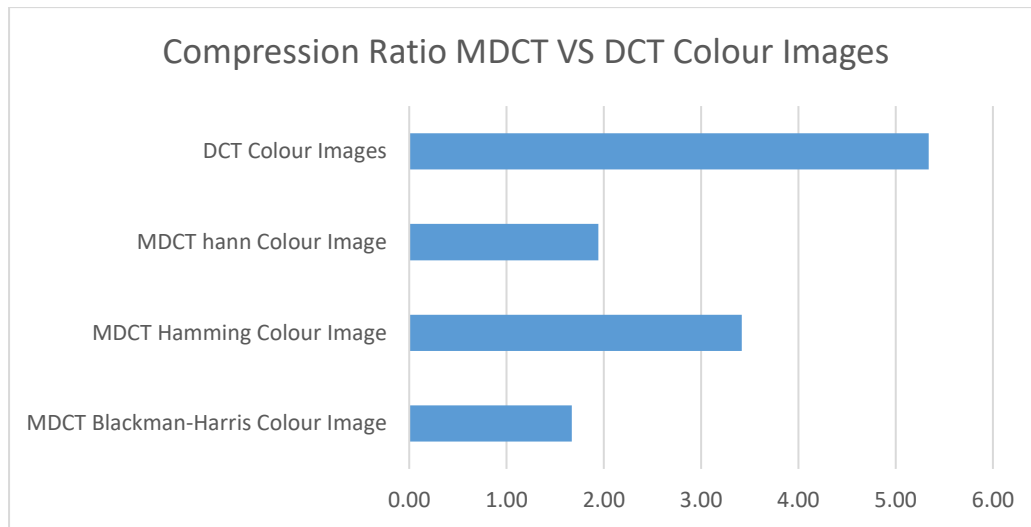
► **1.6714836754012645 (Blackman-Harris)**

➤ **3.418525828041874 (Hamming)**

➤ **1.9459532066624334 (hann)**

DCT Compression Ratio:

➤ **5.340594523945173**



The result shows that the DCT outperforms the MDCT in image compression as higher value indicates better compression ratio.

- **Bits Per Pixel:**

MDCT Bits Per Pixel

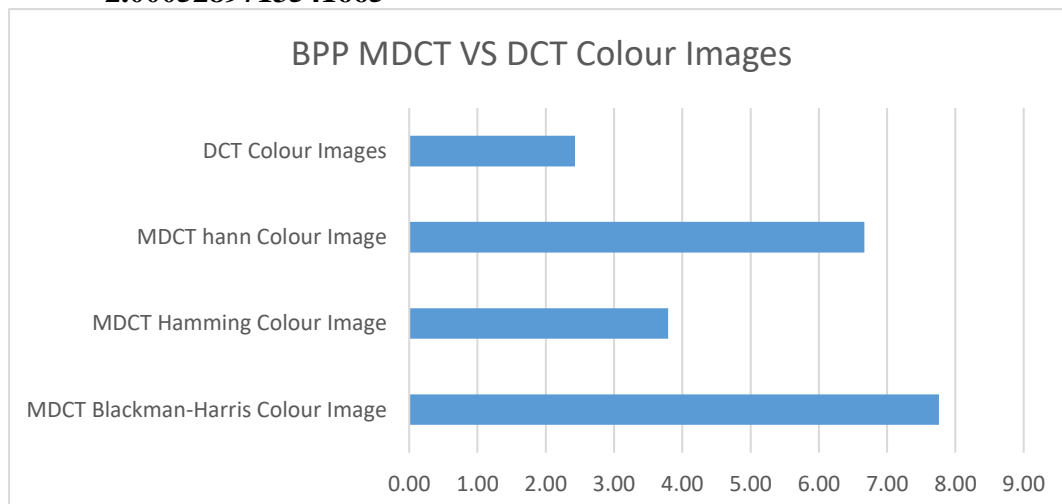
➤ **7.75872802734375 (Blackman-Harris)**

➤ **3.7936197916666665 (Hamming)**

➤ **6.664388020833333 (hann)**

DCT Bits Per Pixel:

➤ **2.0005289713541665**



The result shows that the DCT outperforms the MDCT in compressing the image as lower value indicates better compression of image.

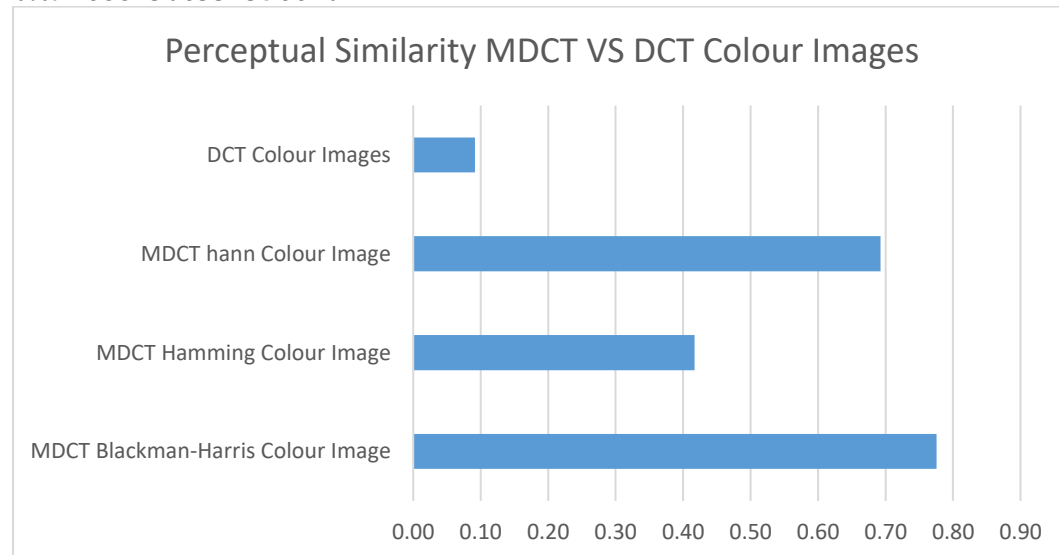
- **Perceptual Similarity:**

MDCT Perceptual Similarity:

- **0.7755568027496338 (Blackman-Harris)**
- **0.4168405532836914 (Hamming)**
- **0.6923218369483948 (hann)**

DCT Perceptual Similarity:

0.09168823063373566702



The result shows that the DCT outperforms the MDCT in compressing the image as lower value indicates better perceptual similarity.

Grayscale Image Results

Following are the results of gray images

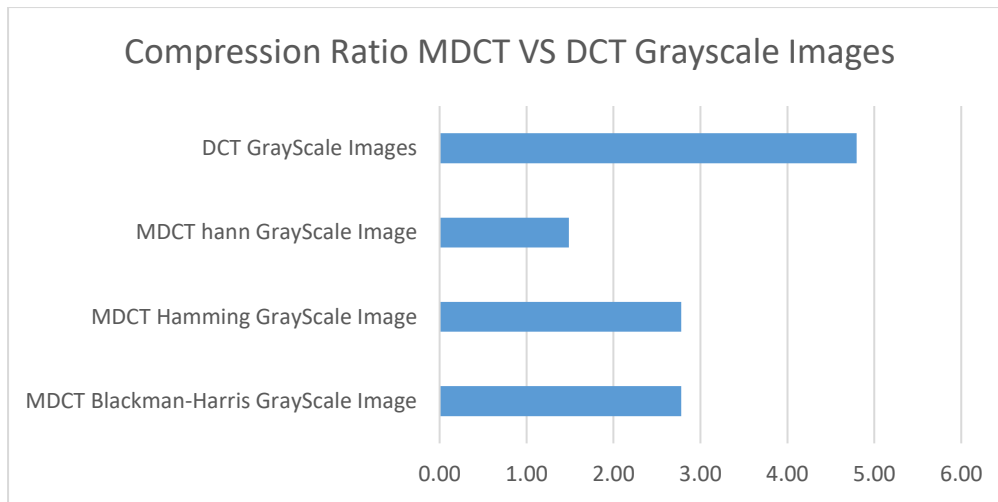
- **Compression Ratio:**

MDCT Compression Ratio:

- **2.7774634789010735 (Blackman-Harris)**
- **2.7774634789010735 (hamming)**
- **1.4856057204048498 (hann)**

DCT Compression Ratio:

- **4.7954773533150465**



The result shows that the DCT outperforms the MDCT in image compression as higher value indicates better compression ratio.

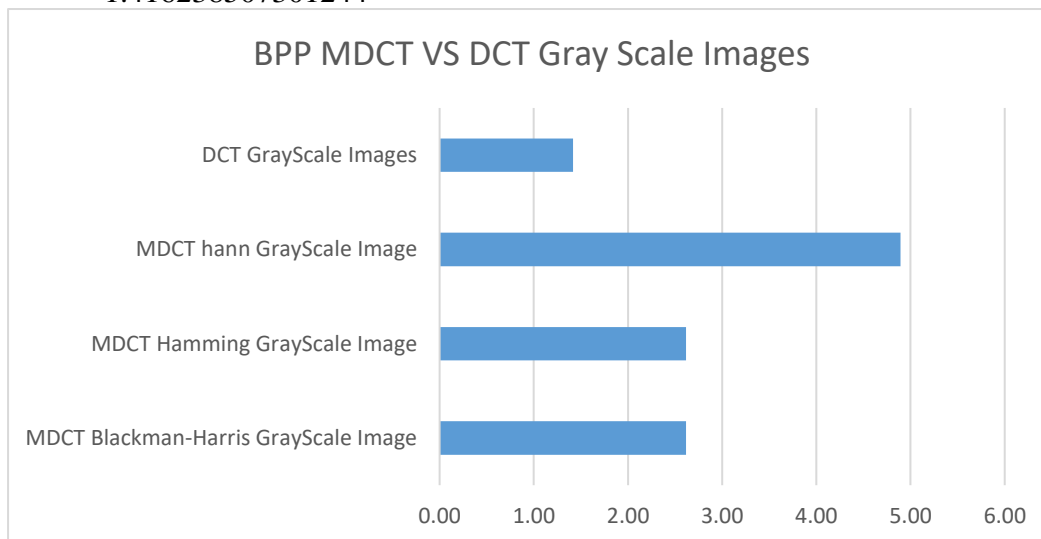
- **Bits Per Pixel:**

MDCT Bits Per Pixel:

- 2.616988912925906 (Blackman-Harris)
- 2.616988912925906 (hamming)
- 4.892678474851271 (hann)

DCT Bits Per Pixel:

- 1.416238507301244



The result shows that the DCT outperforms the MDCT in compressing the image as lower value indicates better compression of image.

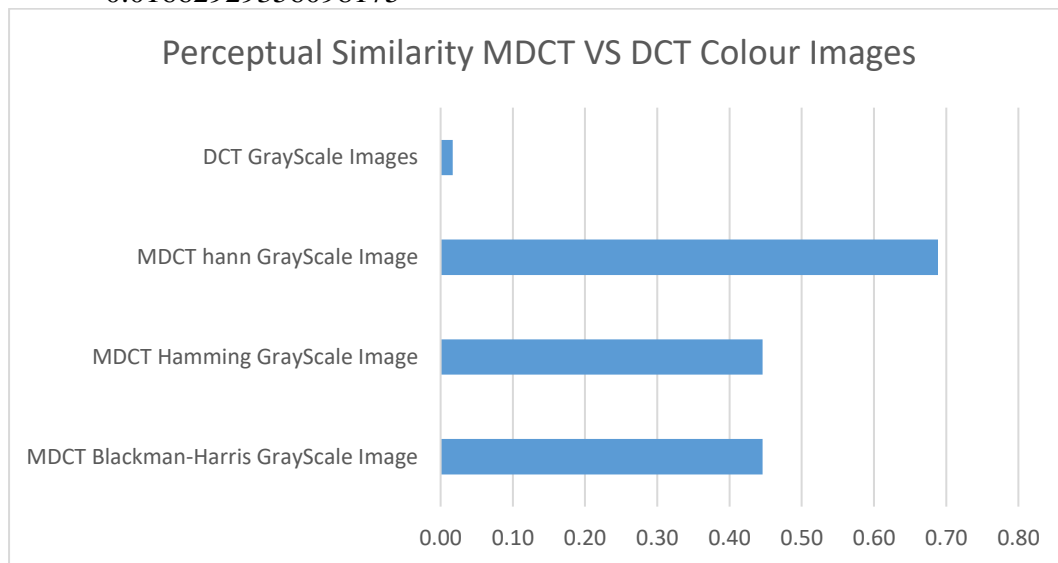
- **Perceptual Similarity:**

MDCT Perceptual Similarity:

- 0.4458668828010559 (Blackman-Harris)
- 0.4458668828010559 (hamming)
- 0.6887151002883911 (hann)

DCT Perceptual Similarity:

➡ 0.01662929356098175



The result shows that the DCT outperforms the MDCT in compressing the image as lower value indicates better perceptual similarity.

5. Conclusion:

After analyzing the results, we concluded that the DCT technique used for image compression is efficient and shows better results than MDCT techniques.