IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 30 September 2025, accepted 2 November 2025, date of publication 12 November 2025, date of current version 1 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3632196

RESEARCH ARTICLE

Simulation of Quantum Discrete Cosine Transform for Grayscale Image Compression Using Qiskit

FELIX MONTALFU[1], SEHAM AL ABDUL WAHID[2], (Graduate Student Member, IEEE), FARAH MOHAMMADI[2], AND ARGHAVAN ASAD[1]

[1]Faculty of Computer Science and Technology, Algoma University, Brampton, ON L6V 1A3, Canada
[2]Electrical and Computer Engineering Department, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada

Corresponding author: Felix Montalfu (fmontalfu@algomau.ca)

This work was supported by Algoma University for the publication fee.

ABSTRACT The Discrete Cosine Transform (DCT) is an integral part of classical image compression, which becomes the basis of the JPEG standard. With advances in quantum computing, there is growing interest in exploring quantum analogues of classical algorithms for such transforms. This paper presents a simulation of the Quantum Discrete Cosine Transform (QDCT) using the Qiskit Python library, applied to standard grayscale images segmented into 8 × 8 and 4 × 4 blocks. The QDCT algorithm is formulated as a unitary operator and evaluated using state-vector simulation. Due to the current constraints of quantum hardware and challenges in circuit synthesis, this study focused on simulation results, as direct quantum circuit measurement leads to amplitude collapse and unusable output. The performance of QDCT is quantitative compared to classical DCT using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and compression ratio metrics. The result of this study reveals the feasibility and current limitations of QDCT-based image compression while providing a reproducible benchmark for future quantum image processing research. This study offers a realistic assessment of the potential and technical limitations of QDCT in the context of emerging quantum technologies.

INDEX TERMS Compression algorithm, discrete cosine transforms, image compression, quantum algorithm, quantum computing, quantum information.

I. INTRODUCTION

With the rapid growth of digital images in communications, entertainment, and scientific domains, efficient compression algorithms have become increasingly crucial. Recently, quantum computing has attracted attention for its potential speedups for specific problems by harnessing superposition and entanglement principles [1]. In 1992, the JPEG specification was published, which implemented the Discrete Cosine Transform (DCT) as the main compression algorithm. JPEG compression splits images into blocks (commonly 8 × 8), applies DCT to each block, and then quantizes the coefficients to control compression rate and visual quality [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

While the Quantum Fourier Transform (QFT) is well-studied, real-valued transforms such as DCT remain relatively unexplored in the quantum context, despite their importance in classical applications, especially in image compression. To address this gap, this paper investigates the simulation of the Quantum Discrete Cosine Transform (QDCT) for image compression using the Qiskit Python library.

Beyond being an academic exercise, QDCT holds potential practical advantages. It provides a foundation for quantum-native compression pipelines, reducing costly data transfer between classical and quantum devices in hybrid architectures. Since it directly operates on quantum states, it enables integration with quantum encryption or communication protocols (e.g., BB84), offering compression and security advantages [3], [4]. Moreover, due to quantum registers scaling logarithmically with input size, QDCT-based approaches may eventually support high-dimensional multimedia data more efficiently than classical DCT, once scalable quantum hardware is available [1], [4], [5], [6]. Compressed quantum states could also serve as low-dimensional inputs for Quantum Machine Learning (QML) models, reducing overhead in emerging applications [7].

This paper benchmarks QDCT against classical DCT using metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and compression ratio. Due to current hardware limitations, the implementation is restricted to statevector simulation, as real quantum hardware and circuit synthesis approaches cause state collapse, leading to unsatisfactory results.

The main contributions of this paper are:

- Quantitative comparison of QDCT and classical DCT on standard test images, reporting PSNR, SSIM, and compression ratio
- Analysis of practical limitations and prospects for quantum image compression
- Formulation and simulation of the QDCT operation for image compression using Qiskit

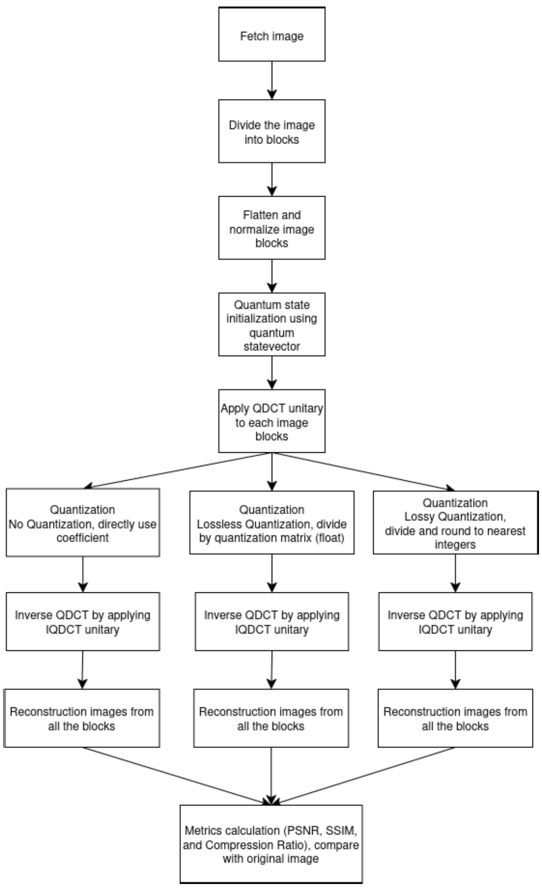


FIGURE 1. QDCT image compression pipeline (statevector simulation).

II. BACKGROUND

In this section, this paper provides the necessary background information to understand the implementation and evaluation

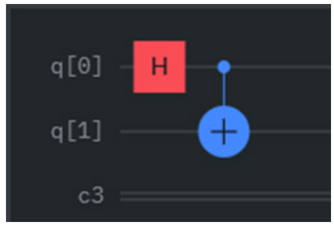


FIGURE 2. Quantum circuit generating entanglement.

of the Quantum Discrete Cosine Transform (QDCT) for image compression. Furthermore, this section will discuss the Discrete Cosine Transform (DCT) and its central role in classical image compression techniques, particularly in the JPEG standard image compression.

A. QUANTUM COMPUTING BASICS

Quantum computing extends classical computation by exploiting superposition, entanglement, and unitary evolution [1], [8], [9]. Unlike classical bits, a qubit can exist in a superposed state

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad |\alpha|^2 + |\beta|^2 = 1, \tag{1}$$

allowing quantum computers to process large state spaces simultaneously. When multiple qubits are combined, the Hilbert space grows exponentially (2^n states for n qubits), and entanglement introduces non-classical correlations. Fig. 2 illustrates a basic two-qubit entangling circuit.

Quantum circuits are built from unitary gates, the quantum analogue of classical logic gates. Standard single-qubit gates include Hadamard (superposition), Pauli-$X/Y/Z$ (axis rotations), and phase rotations, while multi-qubit gates such as CNOT or Toffoli perform controlled operations. Fig. 3 shows an example quantum circuit. Unlike classical computation, measurement collapses superposition to definite outcomes based on the Born rule, making full amplitude readout infeasible on hardware [10].

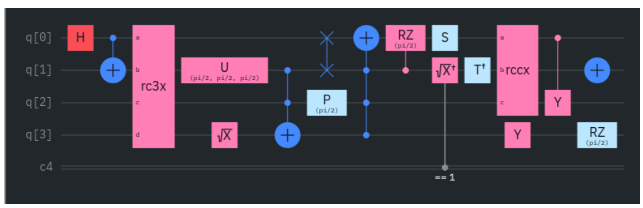


FIGURE 3. Example of a quantum circuit composed of single- and multi-qubit gates.

Given current hardware limitations (low qubit counts, noise, and decoherence), most research relies on quantum simulators such as Qiskit Aer's statevector backend [11]. These simulators enable exploration of quantum algorithms for domains like image processing, including Quantum Discrete Cosine Transform (QDCT), without being restricted by device errors. The relevance to image compression arises because unitary transforms, such as the Fourier or Cosine transform [12], can be represented as quantum circuits and applied to vectorized image blocks, forming the foundation of the methodology used in this work.

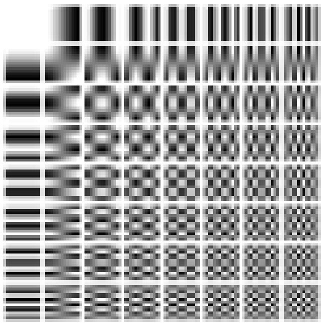


FIGURE 4. DCT coefficient matrix.

B. DISCRETE COSINE TRANSFORM (DCT) AND JPEG COMPRESSION

Discrete Cosine Transform (DCT) is a widely used transformation technique in digital signal and image processing. It was introduced by Ahmed, Natarajan, and Rao in 1974 based on an adaptation of another transform named Karhunen-Loéve Transform (KLT), where it produces an uncorrelated coefficient because the coefficient can be treated independently without loss of compression efficiency [13], [14]. DCT Transforms will take a signal or image from the spatial domain to the frequency domain, with the essential property that most of the signal's energy tends to be concentrated in a small number of low-frequency components [15].

1) DCT IN IMAGE COMPRESSION

In the context of image compression, it uses DCT-2D (Discrete Cosine Transform 2D), which operates on square blocks (commonly using an 8×8 pixel block), transforming pixel values into coefficients that represent the spatial frequency of the pixel values. Since the basis functions of DCT are cosine waves of varying frequency, and the result of the transformation is a set of coefficients, each of the coefficients represents the strength of a particular spatial frequency in the block. Unlike DFT (Discrete Fourier Transform), which produces complex-valued coefficients, DCT produces real-valued output, which makes it computationally efficient and suitable for compression tasks [13], [15].

Most natural images have strong correlations between neighboring pixels, causing DCT to concentrate most of the information in a few low-frequency coefficients (top-left corner of the DCT matrix), while the higher-frequency coefficients (bottom-right) are often small or zero. These low-frequency and high-frequency concentrations are useful since human eyes cannot distinguish high-frequency data even if that particular section is set to zero [15]. In mathematical formulation for DCT-2D, it is defined as:

$$C_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f_{x,y} \cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{2}$$

where:

- $u, v = 0, 1, \ldots, N-1$ are the frequency indices,
- $f(x, y)$ is the pixel value at position (x, y),
- $\alpha(u), \alpha(v)$: normalization factors, where $\alpha(k) = \sqrt{\frac{1}{N}}$ for $k = 0$, $\alpha(k) = \sqrt{\frac{2}{N}}$ for $k > 0$.

After applying the transform to reconstruct the spatial value blocks from their frequency coefficients, it uses the inverse discrete cosine transform 2-dimension (IDCT-2D), and it can be defined in mathematical formulation as:

$$f(x, y) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \alpha(u)\alpha(v)F(u, v) \cos\left[\frac{\pi}{N}\left(x+\frac{1}{2}\right)u\right]\cos\left[\frac{\pi}{N}\left(y+\frac{1}{2}\right)v\right] \tag{3}$$

where:

- $f(x, y)$ are the original pixel intensity at the position (x, y),
- $F(u, v)$ the DCT coefficient multiplies as a cosine basis function of horizontal and vertical frequency,
- $\alpha(u), \alpha(v)$: normalization factors, where $\alpha(k) = \sqrt{\frac{1}{N}}$ for $k = 0$, $\alpha(k) = \sqrt{\frac{2}{N}}$ for $k > 0$,
- $x, y = 0, 1, \ldots, N-1$ is the pixel location in the block,
- $u, v = 0, 1, \ldots, N-1$ is the frequency indices

2) JPEG COMPRESSION STANDARD

The JPEG compression standard, published in 1992, is built around DCT as its primary compression algorithm. The usual JPEG compression pipeline includes several key stages:

- Color space conversion The input image is typically in RGB (Red, Green, Blue) format. JPEG will convert it to YCbCr (Y: Luminance, Cb/Cr: Chrominance), since human eyes are more sensitive to brightness (Y) than color (Cb, Cr). This format allows better compression by reducing color (Cb, Cr) data.
- Chrominance sub-sampling Since human eyes are less sensitive to color, JPEG often reduces the resolution of the Cb/Cr channel, usually with a ratio of 4:2:0 or 4:2:2
- Block splitting The image will be divided into blocks, typically 8×8 pixels. This will help compression since it will limit the computational burden and localize the effect of the quantization.
- 2D DCT Application Each block will undergo DCT, which will convert spatial pixel values into frequency domain coefficients
- Quantization DCT coefficients are divided by a quantization matrix and rounded to the nearest integer. This step introduces lossy compression, prioritizing the retention of perceptually important low-frequency information. In this step, the quality of the JPEG compression is governed by the quantization matrix. If the quantization matrix values are larger, it will reduce the image quality and vice versa

- Zig–zag ordering The DCT coefficients are ordered in a zig-zag pattern while grouping low-frequency values first to help with efficient encoding, as after quantization, many high-frequency coefficient values are zero or near zero.
- Encoding The JPEG standard usually uses Huffman encoding to further compress the values.

For gray-scaled images, color space conversion and chrominance sub-sampling will not be applied, since gray-scale photos do not have a color data channel. Decoding the output image of the JPEG compression can be done by reversing the process [13], [14], [16].

C. QUANTUM DISCRETE COSINE TRANSFORM (QDCT)

While the Quantum Fourier Transform (QFT) is a well-established primitive in quantum algorithms, the QDCT is less explored but holds promise for image and signal processing tasks on quantum hardware [12], [17].

1) DIRECT MATRIX-BASED IMPLEMENTATION

In this research, QDCT is implemented directly as a unitary matrix operation within a quantum circuit using the Qiskit Python library. Unlike other quantum image processing schemes that employ specialized encodings, this study utilizes direct amplitude mapping: small image blocks are flattened into vectors, normalized, and initialized into quantum states via Qiskit's `initialize` instruction. The DCT itself is implemented as a unitary operator, consistent with the requirement that quantum operations must be unitary.

2) PRACTICAL LIMITATION

Because information is encoded directly into quantum amplitudes, retrieving all coefficients on real quantum devices is infeasible due to state collapse upon measurement. Thus, this work relies on statevector simulation to access the full output for benchmarking. Additionally, quantization can produce zero vectors, leading to normalization issues and flat reconstructions. These limitations highlight the current technological gap between theoretical QDCT feasibility and practical realization.

III. RELATED WORK

Image compression and quantum algorithms have been the focus of researchers in recent years, with growing interest in applying classical transforms to quantum computers. This section summarizes prior work on classical DCT/JPEG, the development of QDCT-based schemes, and identifies gaps that motivate this study.

A. CLASSICAL IMAGE COMPRESSION AND DISCRETE COSINE TRANSFORM (DCT)

The Discrete Cosine Transform is a cornerstone of digital signal processing, and it is widely used, especially in image and video compression. Since its introduction and adoption in the JPEG standard [2], [14], [17], DCT has been the foundation of practical image compression for over three decades. The JPEG standard operates by dividing the image into fixed-size blocks (commonly 8×8), applying DCT to each block, quantizing each block's coefficients, and finally encoding the result using Huffman's encoding to reduce redundancy [16]. DCT enables efficient energy compaction, meaning that most information is concentrated in a small subset of coefficients, which allows for significant data reduction with minimal loss in visual quality, as it removes or minimizes high-frequency data from the images or signal.

More recent educational work, such as 'Discrete Cosine Transform JPEG Compression' [15] revisits and modernizes the implementation of DCT, focusing not only on compression performance, but also on pedagogical clarity. This work highlights the robustness and flexibility of DCT-based JPEG, even as digital imaging technology and use cases, such as video compression, have evolved. Moreover, with the advancement of computational technology, the need for efficient and scalable compression methods remains critical, making DCT-based algorithms as relevant as ever.

However, while classical JPEG remains successful, its compression efficiency is fundamentally bounded by classical computation. As image and data volumes continue to grow exponentially [18], researchers have begun to explore whether quantum algorithms could offer performance or security advantages over established classical pipelines.

B. QUANTUM DISCRETE COSINE TRANSFORM (QDCT) AND QUANTUM IMAGE COMPRESSION

With the advancement of quantum computing theory and technologies, especially in quantum simulation, there has been substantial interest in mapping classical signal processing operations such as DCT into the quantum domain. Early theoretical work by Klappenecker and Rotteler [12] established the mathematical formulation of DCT and its fast variants (such as DCT-II, used in JPEG) as quantum circuits, demonstrating that DCT can be expressed as a unitary transformation which is compatible with quantum operations. This foundational step showed feasibility but did not yet provide image-level implementations or benchmarks.

Following Klappenecker and Rotteler's work, numerous studies explored the application of Quantum Discrete Cosine Transform (QDCT) in image compression and encryption. For example, Xiao et al. [4] and Jing et al. [19] developed hybrid quantum compression-encryption schemes, combining QDCT with an additional security layer such as hyper-chaotic Henon maps. These approaches highlight the potential for integrating QDCT into secure communication frameworks, but they focus more on cryptography than on direct comparison to classical compression.

Other studies, such as "Quantum Discrete Cosine Transform for Image Compression" [20] and "Signal and Image Compression Using Quantum Discrete Cosine Transform" [21], focus on the fundamental feasibility and efficiency of QDCT in image compression. *However, their evaluations are* *largely theoretical or limited to abstract simulation scenarios, often without standardized test images or reproducible pipelines.* In the paper "Quantum JPEG" [22], for instance, a quantum implementation of the JPEG standard algorithm is proposed, providing an open-source code, and evaluating quantum compression on simulated platforms. Yet, this work relies on the Quantum Fourier Transform (QFT) rather than QDCT, meaning the direct comparison between classical DCT and its quantum analog remains underexplored.

Another recent study, such as "Advanced Quantum Image Representation and Compression Using A DCT-EFRQI Approach" [5], investigated a more sophisticated hybrid representation of combining QDCT with an advanced quantum image encoding scheme to achieve improved compression ratios and better image fidelity. While valuable, this introduces significant state-preparation overhead, making it less directly comparable to JPEG's block-based workflow.

Taken together, the literature demonstrates that QDCT has been studied in diverse ways—from theoretical formulations to hybrid encryption schemes—but most works either avoid direct benchmarking with classical DCT, rely on alternative transforms like QFT, or assume idealized encoding of quantum images.

C. LIMITATION AND RESEARCH GAP IN QDCT RESEARCH

Despite considerable theoretical advancement, the practical implementation of QDCT-based image compression remains challenging. Many studies [5], [12], [15], [20], [22] use quantum simulation environments (especially statevector simulators) rather than using a real quantum device due to the latter's limitation in qubit counts, gate fidelity, and susceptibility to decoherence, especially in complex quantum circuits. A common limitation in quantum approaches is the issue of measurement collapse: after applying QDCT as a unitary operation, extracting meaningful image information encoded into the qubit amplitude via direct measurement is nontrivial, often resulting in amplitude collapse and loss of data, especially for higher-dimensional images.

Another gap in current research is the direct, empirical comparison of QDCT and classical DCT using real images and practical, interpretable metrics (such as PSNR, SSIM, and compression ratio), particularly in the context of block-based image processing, analogous to the JPEG standard. While theoretical circuit depth and quantum resource estimates are discussed [12], [20], detailed, reproducible code and step-by-step pipelines for quantum image compression are less common, with many studies relying on idealized or abstract assumptions about state preparation and noise. In particular, reproducibility is rarely emphasized: while Roncallo's Quantum JPEG offers open-source implementation, many other QDCT works remain descriptive without accessible implementations.

Furthermore, few works have directly addressed the limitation of amplitude collapse and the impact of using quantum circuit vs. statevector simulators. As a result, a gap remains in practical usage, end-to-end evaluation of QDCT implementation using widely available quantum simulation tools (such as the Qiskit library) and standardized image blocks [23].

D. SUMMARY OF THE CURRENT WORKS

Building upon the extensive body of prior research, this study aims to provide a practical, comparative evaluation of QDCT and classical DCT for block-based image compression. The implementation leverages statevector simulation via Qiskit Aer to avoid amplitude collapse, which resulted in unreadable image data, allowing direct comparison of QDCT outputs to classical DCT on identical input data and compression settings. Unlike previous schemes that may include additional encryption, encoding, or nonstandard representation, this study focuses on a clear, reproducible pipeline that mirrors the classical JPEG workflow as closely as possible within the quantum simulation framework.

In particular, unlike previous approaches, this study provides a direct block-based comparison between QDCT and classical DCT using identical images and quantization matrices, with quantitative evaluation (PSNR, SSIM, compression ratio). By focusing on practical reproducibility in Qiskit and explicitly analyzing limitations (statevector reliance, amplitude collapse), it addresses an unmet need in bridging theory with practice. This contribution is significant because it:

- grounds QDCT in practical benchmarks
- a realistic JPEG-like workflow
- highlights concrete limitations and opportunities for quantum image compression in near-term devices.

IV. METHODOLOGY

This section details the step-by-step process taken to implement, simulate, and compare the Quantum Discrete Cosine Transform (QDCT) and the classical Discrete Cosine Transform (DCT) for image compression. The pipeline is divided into preprocessing, classical and quantum transformation, quantization, reconstruction, and metric evaluation.

A. DATA SELECTION AND PRE-PROCESSING

A single high-resolution 512×512 grayscale image (JPEG Format) is selected for use as a dataset, providing a sufficiently large data domain for block-based analysis. The image is loaded using the OpenCV library and resized to the target dimensions if necessary. Each pixel value is converted to a floating-point format, enabling precise mathematical operations using the NumPy data type. For transform and compression analysis, the image is divided into non-overlapping blocks of size 8×8 or 4×4. There are two different strategies were employed for block extraction:

- **Sequential extraction:** Blocks are taken in raster-scan order, starting from the top-left corner and proceeding row by row until the bottom-right. This method ensures that all blocks are processed, allowing full-image reconstruction. It is mainly used for the primary quantitative evaluation of PSNR, SSIM, and compression ratio.

- **Random extraction:** Blocks are randomly sampled from different image locations using a fixed random seed for reproducibility. This method provides representative samples without reconstructing the full image, and was primarily used during early testing and qualitative visualization, especially when analyzing 4×4 blocks to highlight local variations.

This dual approach allows both complete reconstruction for benchmarking and efficient sampling for exploratory analysis.

B. CLASSICAL DISCRETE COSINE TRANSFORM (DCT) PIPELINE

The 2D-DCT is applied to each image block using the standard DCT-II formula (see section II-B, eq 2), where, in this study, it uses either the SciPy library to perform DCT or a custom matrix DCT multiplication. To simulate JPEG-style compression, this study uses the quantization JPEG standard matrix (or a 4×4 cropped/scaled version). Apart from the size of the quantization matrix, this study explores 3 modes compression, such as:

- No quantization – coefficients are left unmodified.
- Lossless quantization – Division by the quantization matrix; float values are retained (no rounding)
- Lossy quantization – Division and rounding to the nearest integer (this method mimics the JPEG pipeline standard)

To reconstruct the compressed image, inverse 2D-DCT (IDCT) is performed (as described in section II-B, eq 3) to reconstruct the image block.

C. QUANTUM DCT (QDCT) IMPLEMENTATION

The QDCT is implemented using Qiskit and simulated via the statevector simulator backend. Before the QDCT can be implemented, the image block must be flattened and normalized to form a quantum state vector:

$$|\psi\rangle = \frac{1}{||x||}[x_0, x_1, \ldots, x_{N^2-1}]^T \quad (4)$$

Since quantum operation is only an accepted unitary operation, the QDCT operator is constructed as the Kronecker product of two DCT matrices for a 2D transform (e.g., $DCT_4 \otimes DCT_4$), the normalized vector is initialized in a Qiskit QuantumCircuit, the QDCT unitary is applied, and the result is extracted from the statevector. IQDCT is applied to obtain the reconstructed block, which is reshaped to its original dimensions.

D. IMPLEMENTATION DETAILS

This study uses Python 3.X as its code base, and with several libraries/frameworks:

- Qiskit (quantum circuit simulation)
- NumPy (matrix and linear algebra simulation)
- OpenCV (image loading/manipulation)
- Scikit-image (metric computation)
- Matplotlib (visualization)
- SciPy (discrete cosine transform library)

All the code is organized in Jupyter Notebooks for transparency and ease of reproduction.

E. CIRCUIT COMPLEXITY ANALYSIS

To better understand the feasibility of implementing the Quantum Discrete Cosine Transform (QDCT) on quantum hardware, the constructed QDCT and Inverse QDCT (IQDCT) circuits were transpiled into standard quantum gate sets and their depth and gate counts if were measured. Both 4×4 and 8×8 block sizes were analyzed, as they reflect common configurations in classical JPEG compression.

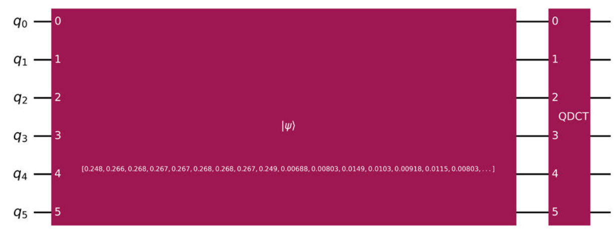


FIGURE 5. 8 × 8 QDCT unitary circuit.

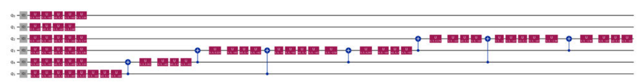


FIGURE 6. 8 × 8 QDCT decomposed circuit.

Fig. 5, fig. 6 and 7 illustrate the transpiled quantum circuit for the QDCT and IQDCT operations. The figures are shown in two formats: a high-level (decomposed) view for clarity, and a decomposed view into native single- and two- qubit gates to reflect the actual resource requirements on real hardware.

TABLE 1. Circuit complexity metrics for QDCT and IQDCT.

Circuit	Depth	Single-Qubit Gates	CNOT Gates	Reset Ops
QDCT-8x8	12,383	18,598	1,925	6
IQDCT-8x8	12,372	18,585	1,925	6
QDCT-4x4	666	1,036	111	4
IQDCT-4x4	650	1,023	111	4

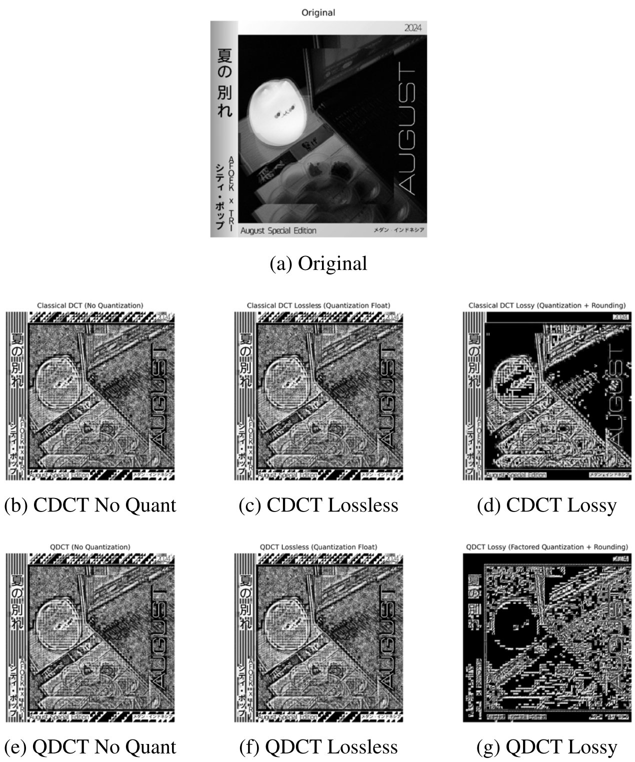
FIGURE 7. 4 × 4 QDCT decomposed circuit.

Table 1 summarizes the circuit complexity metrics. For 8×8 blocks, the QDCT circuit reaches a depth exceeding 12,000 with nearly 10,000 single-qubit gates and 1,925 CNOT gates, which places QDCT well beyond the capability of current Noisy Intermediate-Scale Quantum (NISQ) device.

The 4 × 4 circuits 7 are substantially smaller, with depths around 650 and gate counts near 1,000 single-qubit gates and just over 100 CNOTs, but they remain non-trivial for present-day devices. These result justify the use of statevector simulation rather than execution on bare metal quantum hardware.

F. EVALUATION AND METRICS

Comparison is performed block-by-block, and results are aggregated or visualized using bar charts for Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and compression ratio.



(a) Original

(b) CDCT No Quant (c) CDCT Lossless (d) CDCT Lossy

(e) QDCT No Quant (f) QDCT Lossless (g) QDCT Lossy

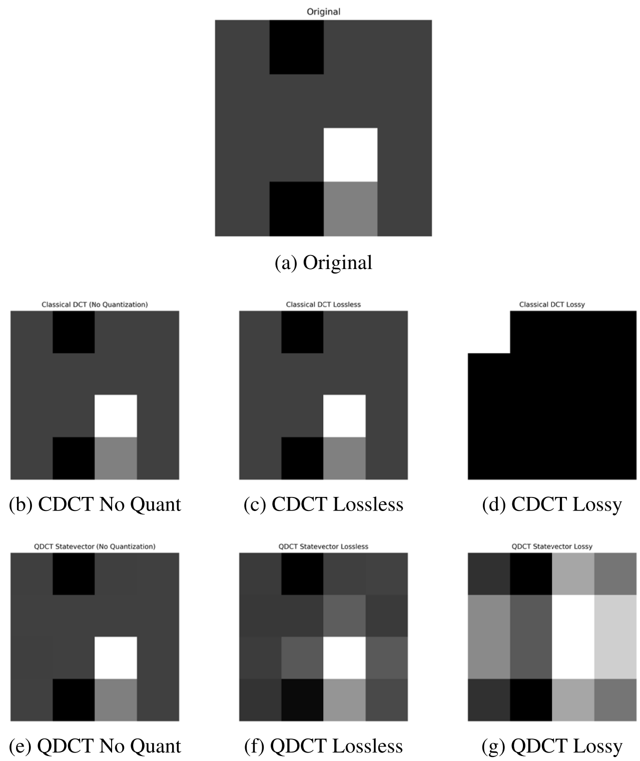
FIGURE 8. Visual comparison for 8 × 8 blocks: Classical DCT (b–d) and QDCT (e–g) under no quantization, lossless, and lossy settings.

V. RESULT AND DISCUSSION

A. VISUAL COMPARISON

Fig. 8 displays the original input image along with all the reconstructed outputs of both classical DCT and QDCT, with approaches for the grayscale image block 8 × 8. Each method is illustrated in three 'modes' compression modes: no quantization, lossless quantization, and lossy quantization:

- **Original image:** The reference image (Fig. 8(a)) is a standard grayscale test image with visually distinct structures and text, providing a challenging yet realistic benchmark for compression and reconstruction.
- **Classical DCT (No Quantization and Lossless Quantization):** Fig. 8(b) and 8(c) confirm that classical DCT achieves nearly identical results to QDCT in the non-lossy cases, as expected from the mathematical equivalence of the transforms.
- **Classical DCT Lossy (Quantization + Rounding):** Fig. 8(d) illustrates that aggressive quantization leads to clear detail loss and blocking effects, similar to the lossy QDCT outcome.
- **QDCT No Quantization and Lossless Quantization:** Fig. 8(e) and 8(f) show that the QDCT, when simulated in statevector mode, reconstructs the image with high fidelity, preserving most visual information with only negligible artifacts.
- **QDCT Lossy (Factored Quantization + Rounding):** In Fig. 8(g), lossy QDCT introduces visible blocking and quantization artifacts. However, the overall image structure remains recognizable.



(a) Original

(b) CDCT No Quant (c) CDCT Lossless (d) CDCT Lossy

(e) QDCT No Quant (f) QDCT Lossless (g) QDCT Lossy

FIGURE 9. Visual comparison for 4 × 4 blocks: Classical DCT (b–d) and QDCT (e–g) under no quantization, lossless, and lossy settings.

For an additional analysis of block size, Fig. 9 presents analogous results for block compression 4 × 4. In all cases, qualitative visual inspection suggests that QDCT simulated via state vector provides results nearly indistinguishable from classical DCT, except when quantization is used.

B. QUANTITATIVE EVALUATION

To complement the visual findings, this study calculated three objective metrics: PSNR, SSIM, and compression ratio for all methods and block sizes. These results are summarized in Table 2 (for 8 × 8 blocks and 4 × 4 blocks) and visualized in Fig. 10.

TABLE 2. QDCT and DCT result.

Image	BS	PSNR	SSIM	CR
QDCT (No Quantization)	8×8	7.946948	0.170360	1.092166
QDCT Lossless (Quantization Float)	8×8	7.946948	0.170360	1.092166
QDCT Lossy (Factored Quantization + Rounding)	8×8	5.675853	0.066131	1.836349
Classical DCT (No Quantization)	8×8	7.946948	0.170360	1.092166
Classical DCT Lossless (Quantization Float)	8×8	7.946948	0.170360	1.092166
Classical DCT Lossy (Quantization + Rounding)	8×8	8.024458	0.208759	1.574378
QDCT (No Quantization)	4×4	60.000000	1.000000	1.000000
QDCT Lossless (Quantization Float)	4×4	25.173758	0.974705	0.933333
QDCT Lossy (Factored Quantization + Rounding)	4×4	9.826521	0.544654	1.000000
Classical DCT (No Quantization)	4×4	60.000000	1.000000	1.000000
Classical DCT Lossless (Quantization Float)	4×4	60.000000	0.974705	1.092166
Classical DCT Lossy (Quantization + Rounding)	4×4	8.085521	-0.011360	14.000000

To further illustrate the trends across methods and block sizes, Figure 10 presents a normalized comparison of PSNR, SSIM, and compression ratio for both 8 × 8 and 4 × 4 block sizes. Normalization enables direct visual comparison by scaling each metric to a [0–1] range, highlighting relative performance rather than absolute magnitudes. The figure highlights relative trends between block sizes: (i) 4×4 blocks consistently achieve higher PSNR and SSIM (reflecting more faithful reconstruction) at the cost of lower compression efficiency, while (ii) 8×8 blocks provide stronger compression ratios but degrade reconstruction quality, especially under lossy quantization. This visualization emphasizes the trade-off between fidelity and compression, which is less immediately apparent from the tabular data. Overall, the combined metrics visualization confirms the expected trade-off between compression efficiency and reconstruction quality, while also emphasizing that QDCT performs comparably to CDCT across block sizes, particularly in the non-lossy scenarios.

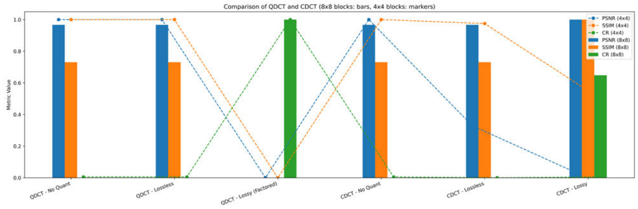


FIGURE 10. Normalized comparison of all metrics for QDCT and classical DCT both 8 × 8 and 4 × 4 blocks.

1) PEAK SIGNAL-TO-NOISE RATIO (PSNR)

In Table 2, the PSNR values for no quantization and lossless quantization cases show a clear dependence on block size. For smaller 4 × 4 blocks, reconstruction accuracy approaches the theoretical maximum ($\approx$ 60 dB), reflecting near-perfect recovery of the original structure. For larger 8 × 8 blocks, PSNR values are notably lower ($\approx$ 7.9 dB), which can be attributed to sensitivity in scaling and normalization when more coefficients are processed simultaneously. This scaling effect amplifies small deviations across the coefficients, producing lower PSNR values even though the reconstructed images remain visually similar to the originals. The trend indicates that block size plays a critical role in the stability of the simulation when using statevector-based QDCT.

2) STRUCTURAL SIMILARITY INDEX (SSIM)

A similar trend is observed in SSIM. For 4 × 4 blocks, SSIM remains close to 1.0, indicating strong structural fidelity in the reconstructions. For 8 × 8 blocks, SSIM drops to $\approx$ 0.17, reflecting higher sensitivity to small coefficient distortions in larger block reconstructions. Since SSIM penalizes luminance and contrast deviations more heavily than PSNR, the metric amplifies these accumulated distortions, resulting in lower reported values. Despite this, the qualitative structure of the reconstructed image is preserved, showing that the numerical degradation in SSIM does not fully reflect visual similarity. This highlights a limitation of applying SSIM directly to simulated quantum reconstructions at larger block sizes.

3) COMPRESSION RATIO

For classical DCT, the lossy quantization mode led to the highest compression ratios, e.g., up to 14 : 1 for classical DCT with 4 × 4 blocks and 1.8:1 for 8 × 8 blocks. QDCT-lossy achieved similarity, but slightly lower compression ratios, demonstrating that quantum approaches can achieve comparable data reduction in principle.

C. DISCUSSION

1) COMPARISON WITH CLASSICAL METHODS

The statevector-based QDCT implementation demonstrates that, when state preparation and measurement are not constrained by hardware, quantum DCT can match classical DCT performances both visually and quantitatively. This is consistent across multiple block sizes and compression levels.

2) LIMITATION OF QUANTUM HARDWARE

Despite promising simulation results, direct hardware implementation remains a key barrier. Attempts to encode image data directly as quantum states and reconstruct via an actual quantum circuit resulted in amplitude collapse, which yielded blank or highly degraded outputs. The success of QDCT in this work relies on the statevector simulation available in Qiskit, which bypasses the difficulties of practical state encoding, decoherence, and measurement.

As a result, actual quantum hardware implementation is not yet feasible for practical compression at this scale; however, this issue can be mitigated using an advanced quantum state tomography – a process in which the quantum state is prepared and measured repeatedly in various bases, allowing the reconstruction of the entire statevector or density matrix. Since, quantum state tomography scales exponentially with the number of qubits, making it infeasible to large-scale images or systems.

3) IMPLICATION AND OUTLOOK

This study validates that QDCT is mathematically equivalent to classical DCT for image compression when simulated ideally. However, major research challenges remain before true quantum advantage can be demonstrated in real-world image processing applications:

- Efficient quantum state preparation: Transforming classical data into quantum states for compression remains an open problem
- Robust quantum state measurement: Amplitude collapse and information loss during measurement are obstacles that must be overcome with better algorithm and method.
- Scalability: The current methodology is limited to small images/blocks due to the exponential scaling of the quantum state space.

VI. CONCLUSION

This study explored the feasibility and performance of implementing the Discrete Cosine Transform (DCT) on quantum computers by simulating a Quantum Discrete Cosine Transform (QDCT) using Qiskit's statevector simulator. The approach was compared to classical DCT (as used in the JPEG standard) using a grayscale image divided into 8×8 and 4×4 blocks.

Experimental results demonstrate that, in the absence of quantum hardware noise and when using the statevector model, QDCT can achieve lossless or near-lossless compression performance comparable to classical DCT, as measured by PSNR and SSIM. In lossy settings, both QDCT and DCT exhibited degradation in reconstruction quality, although the overall structural features of the image were preserved.

However, attempts to encode image data directly into quantum circuits for real-device simulation resulted in amplitude collapse, yielding uninformative (flat) output images. This highlights a significant practical limitation of current quantum hardware and encoding schemes for real-world quantum image compression.

The results confirm that QDCT is mathematically feasible and can be simulated for small-scale images, but real quantum implementation faces substantial challenges, including quantum state preparation, decoherence, and measurement collapse.

FUTURE WORK

There are several avenues for future research to build upon this work:

- Quantum state preparation and tomography: Investigating practical and scalable methods for encoding classical image data into quantum states (beyond amplitude encoding) and reliability extraction using quantum state tomography.
- Hybrid quantum–classical schemes: Developing a hybrid algorithm that combines a quantum subroutine with classical post-processing to mitigate current hardware limitations.
- Compression beyond statevector simulation: Testing QDCT on more realistic quantum simulators (including Noisy Intermediate-Scale Quantum (NISQ) models) and, as hardware improves, on real quantum devices
- Efficient quantum circuit design: Optimizing quantum circuits for the DCT to reduce depth and gate count, making real-device implementation more feasible
- Comparison with other quantum transforms: Benchmarking QDCT against quantum versions of other transforms (such as Quantum Wavelet or Quantum Fourier Transform) for image compression.
- Application-specific benchmarking: Assessing the potential usage of QDCT for other applications, such as medical image storage, satellite imaging, or pattern recognition, where compression efficiency is crucial.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to Dr. Arghavan Asad of Algoma University, Brampton Campus, Canada, for her invaluable guidance and continuous support throughout this research. They also wishes to thank Seham Al Abdul Wahid, Ph.D. (Toronto Metropolitan University), for her expert assistance in editing and proofreading the manuscript. This research was conducted as part of the requirements for the Master of Computer Science Program at Algoma University.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[2] G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben, "JPEG-1 standard 25 years: Past, present, and future reasons for a success," *Proc. SPIE*, vol. 27, no. 4, Aug. 2018, Art. no. 040901.

[3] M. S. Reddy and B. C. Mohan, "Comprehensive analysis of BB84, a quantum key distribution protocol," 2023, *arXiv:2312.05609*.

[4] X.-Z. Li, W.-W. Chen, and Y.-Q. Wang, "Quantum image compression-encryption scheme based on quantum discrete cosine transform," *Int. J. Theor. Phys.*, vol. 57, no. 9, pp. 2904–2919, Sep. 2018.

[5] M. E. Haque, M. Paul, A. Ulhaq, and T. Debnath, "Advanced quantum image representation and compression using a DCT-EFRQI approach," *Sci. Rep.*, vol. 13, no. 1, p. 4129, Mar. 2023.

[6] Z. F. Miller, *Quantum Mechanics for Beginners: With Applications to Quantum Communication and Quantum Computing*. Cham, Switzerland: Springer, 2021.

[7] D. Goldsmith and M. M. H. Mahmud, "Machine learning for quantum computing specialists," 2024, *arXiv:2404.18555*.

[8] N. S. Yanofsky and M. A. Mannucci, *Quantum Computing for Computer Scientist*. New York, NY, USA: Cambridge Univ. Press, 2008.

[9] M. A. Nielsen, *Mathematics of Quantum Computation and Quantum Technology*. Boca Raton, FL, USA: CRC Press, 2008.

[10] S. A. Abdul Wahid, A. Asad, R. Kaur, and F. Mohammadi, "Quantum computing circuit design: A tutorial," in *Proc. Int. Conf. Adv. Sci. Comput. (ICASC)*, Oct. 2024, pp. 1–6.

[11] IBM. *Qiskit*. Accessed: May 2025. [Online]. Available: https://quantum.cloud.ibm.com/docs/en/guides

[12] A. Klappenecker and M. Rotteler, "Discrete cosine transforms on quantum computers," 2001, *arXiv:quant-ph/0111038*.

[13] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York, NY, USA: Van Nostrand, 1993.

[14] (1992). *JPEG*. [Online]. Available: https://jpeg.org/jpeg/index.html

[15] J. John, "Discrete cosine transform in JPEG compression," 2021, *arXiv:2102.06968*.

[16] International Telecommunication Union (ITU), *Information Technology Digital Compression and Coding of Continuous-Tone Still Images Requirements and Guidelines*, CCITT, Geneva, Switzerland, 1992.

[17] K. Ramamohan and H. Ochoa-Domínguez, *Discrete Cosine Transform*, 2nd ed., Boca Raton, FL, USA: CRC Press, 2019.

[18] P. Taylor. (2025). *Volume of Data/Information Created, Captured, Copied, and Consumed Worldwide From 2010 to 2023, With Forecasts From 2024 to 2028*. [Online]. Available: https://www.statista.com/statistics/871513/worldwide-data-created/

[19] J.-Y. Dai, Y. Ma, and N.-R. Zhou, "Quantum multi-image compression-encryption scheme based on quantum discrete cosine transform and 4D hyper-chaotic Henon map," *Quantum Inf. Process.*, vol. 20, no. 7, p. 246, Jul. 2021.

[20] C. Y. Pang, Z. W. Zhou, and G. C. Guo, "Quantum discrete cosine transform for image compression," 2006, *arXiv:quant-ph/0601043*.

[21] C.-Y. Pang, R.-G. Zhou, B.-Q. Hu, W. Hu, and A. El-Rafei, "Signal and image compression using quantum discrete cosine transform," *Inf. Sci.*, vol. 473, pp. 121–141, Jan. 2019.

[22] S. Roncallo, L. Maccone, and C. Macchiavello, "Quantum JPEG," 2023, *arXiv:2306.09323*.

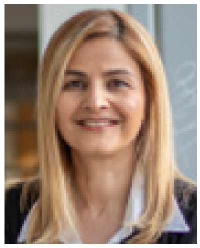
[23] Y. Wilhelm, *Classical and Quantum Computing: With C++ and Java Simulations*. Boca Raton, FL, USA: Taylor & Francis, 2017.

FELIX MONTALFU received the B.Sc. degree in computer science from the Universitas Pelita Harapan, Medan, Indonesia, in 2022. He is currently pursuing the M.Sc. degree in computer science with Algoma University, Brampton Campus, Canada. During his undergraduate studies, he completed a thesis on the comparison of large integer factorization performance between classical and quantum computers, focusing on Shor's algorithm.

He is engaged in research on quantum discrete cosine transform (QDCT) for image compression, and leveraging simulation frameworks, such as Qiskit. He aims to advance the application of quantum technologies in real-world scenarios. He is interested in pursuing further research and doctoral studies in quantum information science. His research interests include quantum computing, quantum information, and quantum neural networks.

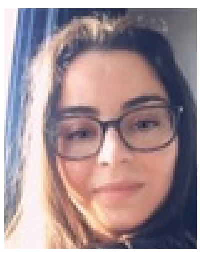
SEHAM AL ABDUL WAHID (Graduate Student Member, IEEE) received the B.Eng. and M.Eng. degrees in biomedical engineering from Toronto Metropolitan University, in 2021 and 2023, respectively, where she is currently pursuing the Ph.D. degree in electrical and computer engineering.

Her research interests include quantum computing, neuromorphic computing, and heterogeneous computing applications and design.



FARAH MOHAMMADI received the B.Sc. and M.Sc. degrees in telecommunication engineering from Iran University of Science and Technology, in 1988 and 1991, respectively, and the Ph.D. degree in electronic engineering from the Institute d'Electronique et Microelectronique du Nord (IEMN), University of Science and Technology of Lille (USTL), France, in 1998.

She was a Faculty Member and a Lecturer with Iran University of Science and Technology, from 1992 to 1993. From 2000 to 2002, she was with Nortel Networks Laboratories, Ottawa, ON, Canada, as a Senior System Architecture with the Fiberoptic Laboratory. Since 2003, she has been with the Electrical and Computer Engineering Department, Toronto Metropolitan University, where she is an Associate Professor. Her research interests include development of electro-thermal test and simulation tools for IC's, design methodologies for RF/microwave components, and applied electromagnetics.



ARGHAVAN ASAD received the first Ph.D. degree in computer science and engineering from Iran University of Science and Technology, Tehran, Iran, in September 2020, and the second Ph.D. degree in electrical and computer engineering from Toronto Metropolitan University, Toronto, ON, Canada in May 2023. She was a Lecturer with Toronto Metropolitan University and Ontario Tech University, ON, from May 2022 to September 2024. In August 2024, she joined the Computer Science Department, Algoma University, as an Assistant Professor. Her research interests include heterogenous AI accelerators, neuromorphic architectures, and quantum computing.