Generalized Quantum Convolution for Multidimensional Data

Mingyoung Jeng *, Alvir Nobel (ID)**, Vinayak Jha, David Levy, Dylan Kneidel, Manu Chaudhary, Ishraq Islam, Muhammad Momin Rahman and Esam El-Araby** (ID)

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA; islam.alvir@ku.edu (A.N.); vinayakjha@ku.edu (V.J.); david.levy@ku.edu (D.L.); dckneidel@ku.edu (D.K.); manu.chaudhary@ku.edu (M.C.); ishraq@ku.edu (I.I.); momin.rahman@ku.edu (M.M.R.); esam@ku.edu (E.E.-A.)
* Correspondence: mingyoungjeng@ku.edu

Abstract: The convolution operation plays a vital role in a wide range of critical algorithms across various domains, such as digital image processing, convolutional neural networks, and quantum machine learning. In existing implementations, particularly in quantum neural networks, convolution operations are usually approximated by the application of filters with data strides that are equal to the filter window sizes. One challenge with these implementations is preserving the spatial and temporal localities of the input features, specifically for data with higher dimensions. In addition, the deep circuits required to perform quantum convolution with a unity stride, especially for multidimensional data, increase the risk of violating decoherence constraints. In this work, we propose depth-optimized circuits for performing generalized multidimensional quantum convolution operations with unity stride targeting applications that process data with high dimensions, such as hyperspectral imagery and remote sensing. We experimentally evaluate and demonstrate the applicability of the proposed techniques by using real-world, high-resolution, multidimensional image data on a state-of-the-art quantum simulator from IBM Quantum.

Keywords: convolution; quantum algorithms; quantum image processing; quantum computing

1. Introduction

Convolution is a common operation that is leveraged in a wide variety of practical applications, such as signal processing [1], image processing [2], and most recently, convolutional neural networks [3]. However, leveraging the widespread utility of convolution operations in quantum algorithms is limited by the lack of a systematic, generalized implementation of quantum convolution. Specifically, contemporary quantum circuits for performing quantum convolution with a given filter are designed on a case-by-case basis [4–8]. In other words, implementing a novel convolution filter on a quantum computer is arduous and time consuming, requiring substantial human effort. Such a workflow is impractical for applications, such as quantum convolutional neural networks, which require a generalized, parameterized quantum circuit to iteratively test thousands of unique filters per training cycle.

In this work, we propose a generalizable algorithm for quantum convolution compatible with amplitude-encoded multidimensional data that is able to implement arbitrary multidimensional filters. Furthermore, our proposed technique implements unity stride, which is essential for capturing the totality of local features in input data. We experimentally verify our technique by applying multiple filters on high-resolution, multidimensional images and report the fidelity of the quantum results against the classically computed expectations. The quantum circuits are implemented on a state-of-the-art quantum simulator from IBM Quantum [9] in both noise-free (as a statevector) and noisy environments. Compared to classical CPU- and GPU-based implementations of convolution, we achieve an exponential improvement in time complexity with respect to data size. Additionally,

when compared to existing quantum implementations, we achieve improved circuit depth complexity when factoring in the data encoding.

The work is structured as follows. In Section 2, we cover important background information and review the related work. In Section 3, we introduce the proposed quantum convolution circuits and provide analyses of the corresponding circuit depth. In Section 4, we present the experimental setup and results, while in Section 5, we provide discussions of the results and comparisons to related work. Finally, in Section 6, we present our conclusions and potential avenues for future explorations and extensions.

2. Background

In this section, we discuss related work pertinent to quantum convolution. Quantum operations that are relevant to convolution, such as quantum data encoding, and quantum shift operation, are also presented.

2.1. Related Work

Classically, the convolution operation is implemented either directly or by leveraging *fast Fourier transform* (FFT). On CPUs, the direct implementation has a time complexity of $\mathcal{O}(N^2)$ [10], where N is the data size, while FFT-based implementation has a time complexity of $\mathcal{O}(N \log N)$ [10]. On GPUs, the FFT-based implementation has a similar $\mathcal{O}(N \log N)$ complexity [11]. It is also common to take advantage of the parallelism offered by GPUs to implement convolution using general matrix multiplications (GEMMs) with $\mathcal{O}(N_F N)$ FLOPS [12,13], where N_F is the filter size.

Techniques for performing quantum convolution have previously been reported [4–8]. However, these techniques only use fixed sizes of filter windows for specific filters, e.g., edge detection [4–8]. We will denote such methods as *fixed-filter* quantum convolution. Reportedly, these methods possess a quadratic circuit depth complexity of $\mathcal{O}(n^2)$ in terms of the number of qubits $n = \lceil \log_2 N \rceil$, where N is the size of the input data [4–8]. Because the shortest execution time of classical convolution is in the order of $\mathcal{O}(N)$ or $\mathcal{O}(2^n)$ [12,13] with respect to data size N, authors of the quantum counterparts often claim a quantum advantage [4]. However, the reported depth complexity of *fixed-filter* quantum convolution does not include the unavoidable overhead of data encoding. Furthermore, there does not exist, to the best of our knowledge, a method for performing generalized, multidimensional quantum convolution.

In reported related work [4–8], data encoding is performed with either the *flexible representation of quantum images* (FRQI) [14] or *novel enhanced quantum representation* (NEQR) [15] methods. In these encoding techniques, positional information is stored in the basis quantum states of n qubits, while color information is stored via angle encoding and basis encoding for FRQI and NEQR, respectively. FRQI and NEQR require a total of $n + 1$ and $n + q$ qubits, respectively, where q is the number of qubits used to represent color values, e.g., $q = 8$ for standard grayscale pixel representation. The reported circuit depth complexities of FRQI and NEQR are $\mathcal{O}(4^n)$ and $\mathcal{O}(qn2^n)$, respectively. When factoring in the depth complexities of either data-encoding technique, it is evident that the referenced *fixed-filter* quantum convolution techniques should be expected to perform worse than classical implementations.

In [16], the authors propose a method of edge detection based on amplitude encoding and the quantum wavelet transform (QWT), which they denote as *quantum Hadamard edge detection* (QHED). Although the work utilizes grayscale two-dimensional images, the QHED technique is highly customized for those data and does not easily scale or generalize to data of higher dimensions, such as colored and/or multispectral images. For example, the quantum discriminator operation in their technique is applied over all qubits in the circuit, without distinguishing between qubits representing each dimension, i.e., image rows or columns. Such a procedure not only forgoes parallelism and increases circuit depth but inhibits the algorithm's ability to be generalized beyond capturing one-dimensional features.

In our proposed work, we achieve an exponential improvement in time complexity compared to classical implementations of convolution with respect to data size. Additionally, when compared to existing quantum convolution implementations, we achieve improved circuit depth complexity when factoring in the data encoding. The contribution of our work is analyzed, experimentally verified, and discussed in detail in Section 5.

2.2. Classical to Quantum (C2Q)

Our method of quantum convolution leverages *amplitude encoding*, which encodes N data values directly in the complex probability amplitudes $c_i \in \mathbb{C}$ of the positional basis state $|i\rangle$ for an n-qubit state $|\psi\rangle$, where $n = \lceil \log_2 N \rceil$ and $0 \leq i < n$, see (1):

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle : c_i \in \mathbb{C}.$$ (1)

We use the *classical-to-quantum* (C2Q) [17] data-encoding technique to encode the amplitude encoded state $|\psi_0\rangle$ from the ground state $|0\rangle^{\otimes n}$, see Figure 1 and (2). The C2Q operation U_{C2Q} has a circuit depth complexity of $\mathcal{O}(2^n)$, a quadratic and linear improvement over FRQI and NEQR, respectively:

$$\begin{aligned}
U_{\text{C2Q}}|0\rangle^{\otimes n} &= |\psi_0\rangle \\
U_{\text{C2Q}} &= \begin{bmatrix} |\psi_0\rangle & |\times\rangle & \cdots & |\times\rangle \end{bmatrix}, \text{where} \\
|\times\rangle &= \text{"don't care"}
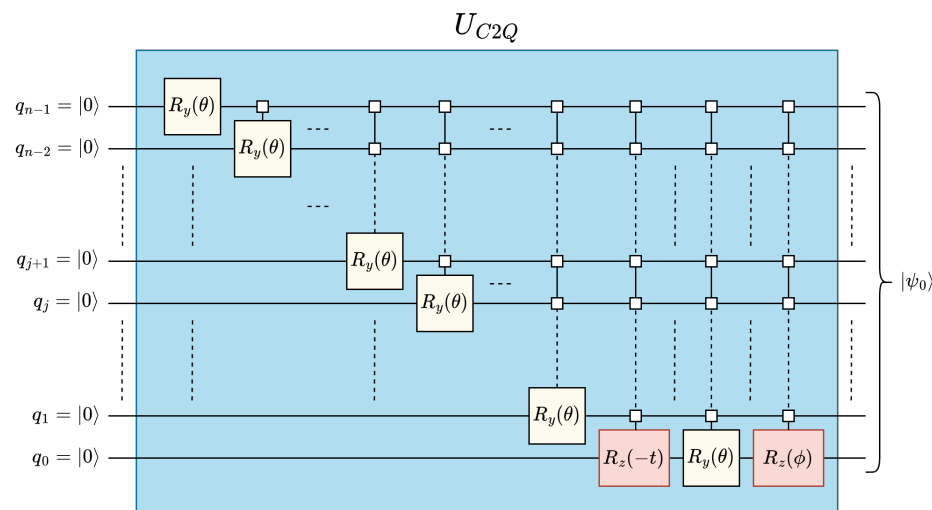\end{aligned}$$ (2)

Figure 1. Quantum circuit for classical-to-quantum (C2Q) arbitrary state synthesis [17].

2.3. Quantum Shift Operation

A fundamental operation for quantum convolution is the *quantum shift operation*, denoted in this work as U_{shift}^k, which shifts the basis states of the state vector by k positions when applied to an m-qubit state $|\psi\rangle$, see (3). The quantum shift operation is critical for performing the cyclic rotations needed to prepare strided windows when performing convolution. It is also common for the operation to be described as a *quantum incrementer* when $k > 0$, see Figure 2a, and a *quantum decrementer* when $k < 0$ [16,18], see Figure 2b:

$$U_{\text{shift}}^k|\psi\rangle = \sum_{i=0}^{2^m-1} c_i|j\rangle, \text{where } j = (i-k) \bmod 2^m$$ (3)

(**a**) Quantum incrementer by k
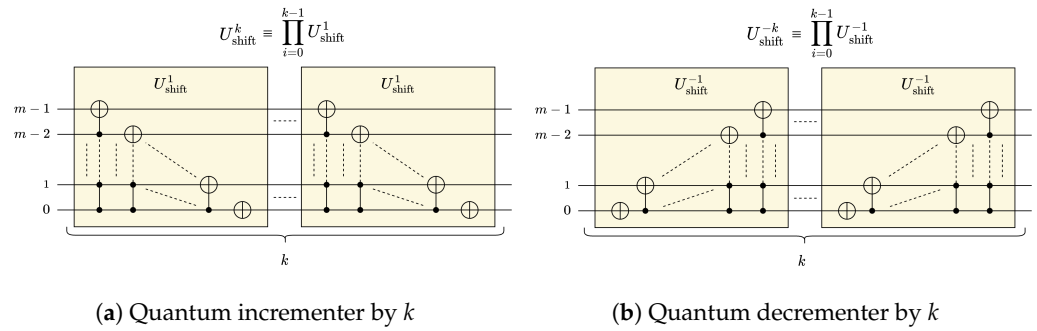
(**b**) Quantum decrementer by k

Figure 2. Quantum shift operation using quantum incrementers/decrementers.

3. Materials and Methods

In general, a convolution operation can be performed using a sequence of shift and multiply-and-accumulate operations. In our proposed methods, we implement the generalized convolution operations as follows:

1. *Shift*: Auxiliary filter qubits and controlled quantum decrementers are used to create shifted (unity-strided) replicas of input data.
2. *Multiply-and-accumulate*: Arbitrary state synthesis and classical-to-quantum (C2Q) encoding are applied to create generic multidimensional filters.
3. *Data rearrangement*: Quantum permutation operations are applied to restructure the fragmented data into one contiguous output datum.

In Section 3.1, we present our quantum convolution technique in detail for one-dimensional data. In the following sections, we illustrate optimizations to improve circuit depth and generalize our method for multidimensional data. For evaluating our proposed methods, we used real-world, high-resolution, black-and-white (B/W) and RGB images, ranging in a resolution from (8×8) pixels to (512×512) pixels and $(8 \times 8 \times 3)$ pixels to $(512 \times 512 \times 3)$ pixels, respectively. We also performed experiments on 1-D real-world audio data and 3-D real-world hyperspectral data to demonstrate our method's applicability to data and filters of any dimensionality. Further details about our experimental setup and dataset can be found in Section 4.

3.1. Quantum Convolution for One-Dimensional Data

The proposed structure of quantum convolution for one-dimensional (1-D) data is shown in Figure 3. The following sections show the details of the five steps of the convolution operation procedure to transform the initial encoded data $|\psi_0\rangle$ to the final state $|\psi_5\rangle$, see Figure 3.
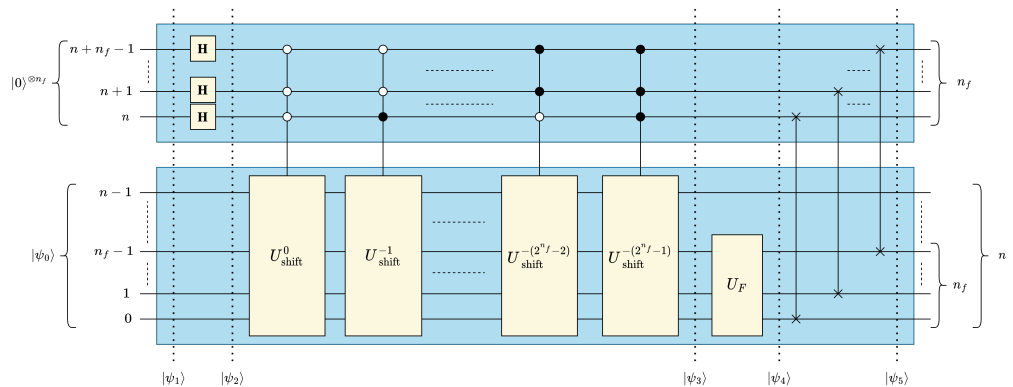

Figure 3. The 1-D quantum convolution circuit.

3.1.1. Shift Operation

To perform convolution with unity stride with a filter of size N_f terms, N_f replicas of the input data must be made, strided for $0 \leq k < N_f$. To store these replicas, we add $n_f = \lceil \log_2 N_f \rceil$ auxiliary qubits, which we denote as "filter qubits", to the most significant positions of the initial quantum state $|\psi_0\rangle$, see (4) and Figure 3:

$$|\psi_1\rangle = |0\rangle^{\otimes n_f} \otimes |\psi_0\rangle = \begin{bmatrix} |\psi_0\rangle \\ 0 \\ \vdots \\ 0 \end{bmatrix} \updownarrow 2^n \Bigg\} 2^{n+n_f} \tag{4}$$

Placing the filter qubits in superposition using Hadamard (H) gates creates 2^{n_f} identical replicas of the initial data $|\psi_0\rangle$, as shown in (5):

$$|\psi_2\rangle = \left(H^{\otimes n_f} \otimes I^{\otimes n} \right)|\psi_1\rangle = \frac{1}{\sqrt{2^{n_f}}} \begin{bmatrix} |\psi_0\rangle \\ \vdots \\ |\psi_0\rangle \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \vdots \\ \updownarrow 2^n \end{matrix} \Bigg\} 2^{n+n_f} \tag{5}$$

Finally, multiplexed quantum shift operations can be used to generate the strided replicas, see (6):

$$|\psi_3\rangle = U_{\text{mux}}|\psi_2\rangle = \frac{1}{\sqrt{2^{n_f}}} \begin{bmatrix} U_{\text{shift}}^{0}|\psi_0\rangle \\ \vdots \\ U_{\text{shift}}^{-(2^{n_f}-1)}|\psi_0\rangle \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \vdots \\ \updownarrow 2^n \end{matrix} \Bigg\} 2^{n+n_f}, \text{ where}$$

$$U_{\text{mux}} = \begin{bmatrix} U_{\text{shift}}^{0} & & \\ & \ddots & \\ & & U_{\text{shift}}^{-(2^{n_f}-1)} \end{bmatrix} \tag{6}$$

3.1.2. Multiply-and-Accumulate Operation

For the traditional convolution operation, applying a filter $F \in \mathbb{R}^{N_f}$ to an array of data $W \in \mathbb{R}^{N_f}$ produces a scalar output $x \in \mathbb{R}$, which can be expressed as $x = F^T W$. In the quantum domain, we can represent an array of data as the partial state $|\phi\rangle$ and the normalized filter as $|F\rangle$. Accordingly, the output state can be expressed as shown in (7):

$$|\psi_{\text{out}}\rangle = \sum_{i=0}^{2^n-1} \langle F|\phi_i\rangle \cdot |i\rangle, \text{ where}$$

$$|\phi_i\rangle = \sum_{j=0}^{2^{n_f}-1} \langle k'|\psi_3\rangle \cdot |k'\rangle, \text{ and } k' = (2^{n_f} \cdot i) + j \tag{7}$$

To calculate $|\psi_4\rangle$ from $|\psi_3\rangle$, it is necessary to embed $\langle F|$ into a unitary operation U_F as shown in (8). Since $\langle F|$ is a normalized row vector, we can define U_F as a matrix such that its first row is $\langle F|$ and the remaining rows are arbitrarily determined to preserve the unitariness of U_F such that $U_F^\dagger U_F = U_F U_F^\dagger = I^{\otimes n_f}$. From (2), we can realize U_F using an inverse C2Q operation, see (8):

$$|\psi_4\rangle = \left(I^{\otimes n} \otimes U_F\right)|\psi_3\rangle = \begin{bmatrix} U_F|\phi_0\rangle \\ \vdots \\ U_F|\phi_{2^n-1}\rangle \end{bmatrix} \begin{matrix} \updownarrow 2^{n_f} \\ \vdots \\ \updownarrow 2^{n_f} \end{matrix} \Bigg] 2^{n+n_f}, \text{ where}$$

$$U_F = \begin{bmatrix} \langle F| \\ \langle \times| \\ \vdots \\ \langle \times| \end{bmatrix} 2^{n_f} = U_{\text{C2Q}}^{\dagger}$$

(8)

3.1.3. Data Rearrangement

As of $|\psi_4\rangle$, the desired values of $|\psi_{\text{out}}\rangle$ are fragmented among undesired/extraneous values, which we denote using the symbol "$\times$". We apply SWAP permutations to rearrange and coalesce our desired values to be contiguous in the final statevector $|\psi_5\rangle$, see (9) and Figure 3:

$$|\psi_5\rangle = U_{\text{SWAP}}^{1-D}|\psi_4\rangle$$

$$= \begin{bmatrix} \langle F|\phi_0\rangle \\ \vdots \\ \langle F|\phi_{2^n-1}\rangle \\ \times \\ \vdots \\ \times \end{bmatrix} \begin{matrix} \Big] 2^n \\ \ \end{matrix} \Bigg] 2^{n+n_f}$$

$$= \begin{bmatrix} |\psi_{out}\rangle \\ \times \\ \vdots \\ \times \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \ \end{matrix} \Bigg] 2^{n+n_f}, \text{ where}$$

(9)

$$U_{\text{SWAP}}^{1-D} = \prod_{j=n_f-1}^{0} \left(I^{\otimes(n_f-1-j)} \otimes \text{SWAP}(j, j+n) \otimes I^{\otimes j} \right)$$

3.1.4. Circuit Depth Analysis of 1-D Quantum Convolution

When considering the circuit depth complexity of the proposed 1-D quantum convolution technique, it is evident from Figure 3 that the operations described by (5) and (9) are performed using parallel Hadamard and SWAP operations, respectively, and thus are of constant depth complexity, i.e., $\mathcal{O}(1)$. In contrast, the U_{mux} and U_F operations incur the largest circuit depth, as they are both serial operations that scale with the data size and/or filter size, see Figure 3.

For the implementation of U_{mux} in Figure 3, there are a total of 2^{n_f} controlled quantum shift operations, where the i-th shift operation is a quantum decrementer $U_{\text{shift}}^{-i} = \left(U_{\text{shift}}^{-1}\right)^i$. Since all the U_{shift}^{-1} operations are performed in series, the circuit depth of U_{mux} depends on the total number of unity quantum decrementers, $N_{U_{\text{shift}}^{-1}}$, see (10):

$$N_{U_{\text{shift}}^{-1}}(n_f) = \sum_{i=0}^{2^{n_f}-1} i = \frac{2^{n_f}(2^{n_f}-1)}{2} = \frac{4^{n_f}}{2} - 2^{n_f-1}$$

(10)

As shown in Figure 2b, each quantum decrementer U_{shift}^{-1} acting on m qubits can be realized using m multi-controlled CNOT (MCX) gates, where the i-th MCX gate is controlled by i qubits and $0 \le i < m$. Accordingly, for each quantum decrementer U_{shift}^{-1} that is controlled by c qubits, its i-th MCX gate is controlled by a total of $i + c$ qubits.

Therefore, the depth of the quantum decrementer circuits can be expressed in terms of the MCX gate count as shown in (11):

$$\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c) = \sum_{i=c}^{m+c-1} \mathcal{D}_{\text{MCX}}(i) \tag{11}$$

The depth of an MCX gate with a total of m qubits can be expressed with a linear function in terms of fundamental single-qubit rotation gates and CNOT gates [19] as shown in (12), where α represents the first-order coefficient and β represents the constant bias term. Thus, the depth complexity of U_{shift}^{-1} can be expressed as shown in (13):

$$\mathcal{D}_{\text{MCX}}(m) = \alpha m + \beta : \alpha, \beta \in \mathbb{R} \tag{12}$$

$$
\begin{aligned}
\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c) &= \sum_{i=0}^{m-1} \alpha(i + c) + \beta \\
&= \frac{\alpha}{2} m^2 + \left(\alpha \left(c - \frac{1}{2} \right) + \beta \right) m \\
&= \mathcal{O}\left(m^2 \right)
\end{aligned}
\tag{13}
$$

To derive the circuit depth complexity of U_{mux}, we leverage the definitions of $N_{U_{\text{shift}}^{-1}}(n_f)$ and $\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c)$ as shown in (14), where $m = n$ and $c = n_f$:

$$
\begin{aligned}
\mathcal{D}_{U_{\text{mux}}}(n, n_f) &= N_{U_{\text{shift}}^{-1}}(n_f) \cdot \mathcal{D}_{U_{\text{shift}}^{-1}}(n, n_f) \\
&= \left(\frac{4^{n_f}}{2} - 2^{n_f - 1} \right) \cdot \left(\frac{\alpha}{2} n^2 + \left(\alpha \left(n_f - \frac{1}{2} \right) + \beta \right) n \right) \\
&= \left(4^{n_f - 1} - 2^{n_f - 2} \right) \cdot \left(\alpha n^2 + 2\alpha n_f n - (\alpha - 2\beta)n \right) \\
&= \mathcal{O}\left(4^{n_f} n^2 + 4^{n_f} n_f n \right)
\end{aligned}
\tag{14}
$$

As discussed in Section 3.1.2, we implement the filter operation U_F by leveraging the C2Q arbitrary synthesis operation [17]. Although C2Q incurs a circuit depth of exponential complexity in terms of fundamental quantum gates, as shown in (15), U_F is only applied to n_f qubits, a small subset of the total number of qubits, which somewhat mitigates the circuit depth. Furthermore, in most practical scenarios, the dimensions of the filter are typically much smaller than the dimensions of the input data, i.e., $n_f \ll n$. As a result, U_F should not incur overly large circuit depth relative to other circuit components, e.g., U_{mux}. Altogether, the overall circuit depth complexity of the 1-D quantum convolution operation can be expressed according to (16):

$$\mathcal{D}_{U_F}(n_f) = \mathcal{O}(2^{n_f}) \tag{15}$$

$$\mathcal{D}_{\text{1-D conv}}(n, n_f) = \mathcal{O}\left(4^{n_f} n^2 + 4^{n_f} n_f n + 2^{n_f} \right), \text{ where } n \gg n_f \tag{16}$$

3.2. Depth-Optimized 1-D Quantum Convolution

In Figure 4, we present an optimized implementation of U_{mux} that greatly reduces the circuit depth.

In Section 3.1, we implemented U_{mux} with 2^{n_f} controlled quantum decrementers U_{shift}^{-k}, where $0 \leq k < 2^{n_f}$. We can represent each k in binary notation, as shown in (17), to express U_{shift}^{-k} as a sequence of controlled shift operations by powers of 2. As shown in (18), we can

denote such operations with the notation $U_{\text{shift}}^{-b_j 2^j}(n)$, where $0 \leq j < n_f$, and (n) reflects that the shift operation is applied to an n-qubit state.

$$k = \left(b_{n_f-1} b_{n_f-2} \cdots b_j \cdots b_1 b_0 \right)_2 = \sum_{j=0}^{n_f-1} b_j 2^j : b_j \in \{0,1\} \tag{17}$$

$$U_{\text{shift}}^{-k}(n) = U_{\text{shift}}^{-\sum_{j=0}^{n_f-1} b_j 2^j}(n) = \prod_{j=0}^{n_f-1} U_{\text{shift}}^{-b_j 2^j}(n) \tag{18}$$
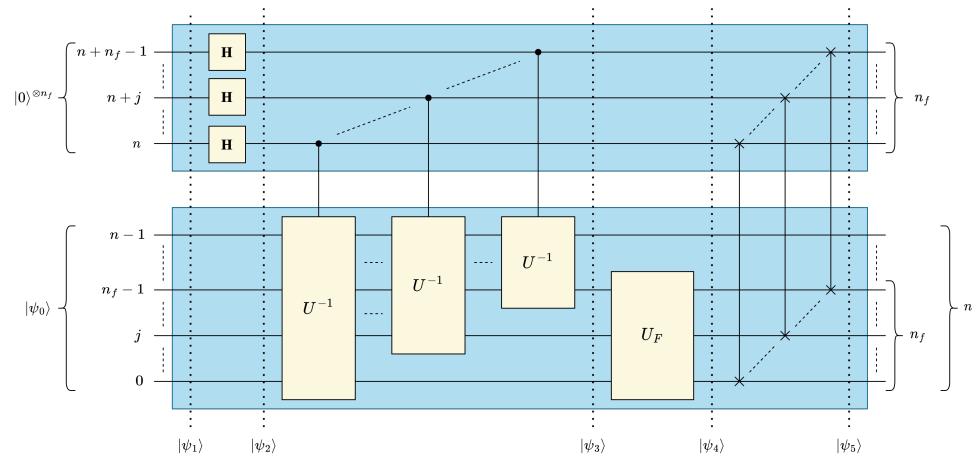

Figure 4. Depth-optimized 1-D quantum convolution circuit.

The binary decomposition of the uniformly controlled U_{shift}^{-k} operations is conducive to several circuit depth optimizations. As shown in (19), the value of b_j is dependent only on the state of the j-th filter qubit q_{n+j}. In other words, each $U_{\text{shift}}^{-2^j}(n)$ can only be controlled by one qubit q_{n+j}, independently from the other control qubits. Accordingly, it is possible to coalesce the multiplexed $U_{\text{shift}}^{-2^j}(n)$ operations across the k control indices. The resultant implementation of U_{mux}, therefore, becomes a sequence of 2^{n_f} single-controlled $U_{\text{shift}}^{-2^j}(n)$ operations, where $0 \leq j < n_f$, which comparatively has a smaller circuit depth by a factor of 2^{n_f}. Furthermore, each $U_{\text{shift}}^{-2^j}(n)$ operation can be implemented using a single $U_{\text{shift}}^{-1}(n-j)$ operation in lieu of sequential $U_{\text{shift}}^{-1}(n)$ operations, see (20) and Figure 4, further reducing the depth by a factor of 2^j per operation:

$$b_j = \begin{cases} 1, & |q_{n+j}\rangle = |1\rangle \\ 0, & \text{otherwise} \end{cases}, \quad \forall k \in [0, 2^{n_f}] \tag{19}$$

$$U_{\text{shift}}^{-2^j}(n) \equiv \prod_{2^j} U_{\text{shift}}^{-1}(n) = U_{\text{shift}}^{-1}(n-j) \otimes I^{\otimes j} \tag{20}$$

Circuit Depth Analysis of Optimized 1-D Quantum Convolution

With the aforementioned optimizations, the depth of the updated U_{mux} operation can be expressed in terms of $\mathcal{D}_{U_{\text{shift}}^{-1}}(m,c)$ as described by (21), where $m = n - j$ and $c = 1$ for all $0 \leq j < n_f$. In comparison with the depth complexity of the unoptimized U_{mux}, see (14), the dominant term remains quadratic, i.e., n^2, in terms of the data qubits n. However, its coefficient is improved exponentially, from 4^{n_f} to n_f, see (14) and (21). Note that a cubic term n_f^3 in terms of the number of filter qubits is introduced in the optimized U_{mux} implementation, see (21). However, when considering the total depth for the optimized 1-D quantum convolution circuit $\mathcal{D}_{\text{1-D conv}}^{\text{opt}}$, the n_f^3 term becomes negligible because of U_F,

whose complexity $\mathcal{O}(2^{n_f})$ is exponential in terms of the number of filter qubits, see (15), (21), and (22):

$$
\begin{aligned}
\mathcal{D}^{\text{opt}}_{U_{\text{mux}}}(n, n_f) &= \sum_{j=0}^{n_f-1} \mathcal{D}_{U^{-1}_{\text{shift}}}(n-j, 1) = \sum_{j=0}^{n_f-1} \left[\frac{\alpha}{2}(n-j)^2 + \left(\frac{\alpha}{2}+\beta\right)(n-j) \right] \\
&= \frac{\alpha}{2} n_f n^2 - \frac{\alpha}{2} n_f^2 n + (\alpha+\beta) n_f n + \frac{\alpha}{6} n_f^3 - \frac{\alpha+\beta}{2} n_f^2 + \frac{2\alpha+3\beta}{6} n_f \\
&= \mathcal{O}\left(n_f n^2 - n_f^2 n + n_f^3 \right), \text{ where } n \gg n_f
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
\mathcal{D}^{\text{opt}}_{\text{1-D conv}}(n, n_f) &= \mathcal{O}\left(n_f n^2 - n_f^2 n + n_f^3 + 2^{n_f} \right) \\
&= \mathcal{O}\left(n_f n^2 - n_f^2 n + 2^{n_f} \right), \text{ where } n \gg n_f
\end{aligned}
\tag{22}
$$

3.3. Generalized Quantum Convolution for Multidimensional Data and Filters

In this section, we present the quantum circuit of our proposed quantum convolution technique generalized for multidimensional data and filters. Although quantum statevectors are one-dimensional, it is possible to map multidimensional data to a 1-D vector in either *row-* or *column*-major order. In this work, we represent multidimensional input data and convolutional filters in a quantum circuit in *column-major order*. In other words, for d-dimensional data of size $(N_0 \times \cdots \times N_i \cdots \times N_{d-1})$ data values, the positional information of the i-th dimension is encoded in the $\sum_{j=0}^{i-1} n_j$ to $\left(\sum_{j=0}^{i} n_j\right) - 1$ qubits, where $n_i = \lceil \log_2 N_i \rceil$. Using this representation, the optimized 1-D quantum convolution circuit shown in Figure 4 can be generalized for d dimensions by "*stacking*" d 1-D circuits as shown in Figure 5.
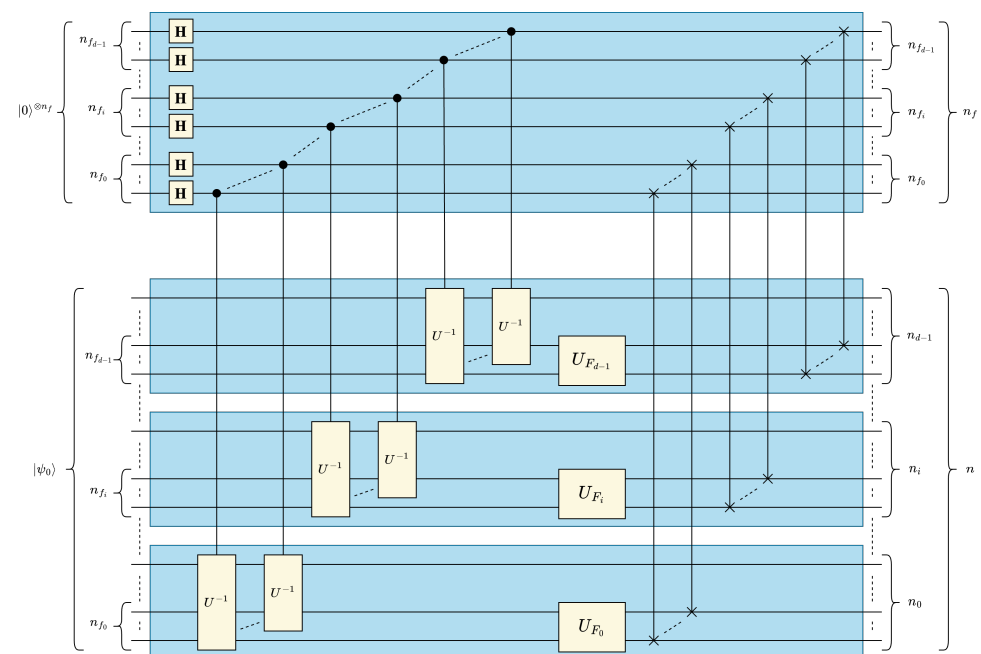
Figure 5. Multidimensional quantum convolution circuit with distributed/stacked 1-D filters.

The "stacked" quantum circuit in Figure 5 is based on the assumption that the overall (lumped) d-dimensional filter operator U_F is separable and decomposable into d one-dimensional filters U_{F_i} for $0 \leq i < d$. However, it would be more practically useful to generalize our multidimensional quantum convolution technique independently from the separability/decomposability of U_F. For this purpose, the identity in (23), which could be easily derived from either Figure 3 or Figure 4 for 1-D convolution, can be leveraged and

generalized for multidimensional convolution circuits, see (24). The identity in (24) allows us to reverse the order of multiply-and-accumulate and data rearrangement steps and, therefore, generate one generic lumped U_F that acts on the contiguous $n_f = \sum_{i=0}^{d-1}(n_{f_i})$ filter qubits, where n_{f_i} is the number of qubits representing the filter dimension i for $0 \leq i < d$, see Figure 6. U_F can be derived based on the given arbitrary multidimensional filter F using the method discussed in Section 3.1.2 when F is represented as a normalized 1-D vector $|F\rangle$ in a column major ordering:

$$U_{\text{SWAP}}^{1-D} \cdot \left(I^{\otimes n} \otimes U_F\right) = \left(U_F \otimes I^{\otimes n}\right) \cdot U_{\text{SWAP}}^{1-D} \tag{23}$$

$$U_{\text{SWAP}}^{d-D} \cdot \left(I^{\otimes n_f} \otimes \left(\bigotimes_{i=d-1}^{0} \left[I^{\otimes(n_i - n_{f_i})} \otimes U_{F_i}\right]\right)\right) = \left(\left(\bigotimes_{i=d-1}^{0} \left[U_{F_i}\right]\right) \otimes I^{\otimes n}\right) \cdot U_{\text{SWAP}}^{d-D}$$

$$= \left(U_F \otimes I^{\otimes n}\right) \cdot U_{\text{SWAP}}^{d-D}, \text{ where}$$

$$U_F = \bigotimes_{i=d-1}^{0} \left[U_{F_i}\right] \equiv U_{F_{d-1}} \otimes U_{F_{d-2}} \otimes \cdots \otimes U_{F_1} \otimes U_{F_0}, \tag{24}$$

$$U_{\text{SWAP}}^{d-D} = \prod_{i=d-1}^{0} \prod_{j=n_{f_i}-1}^{0} \left(I^{\otimes\left(n_f - 1 - j - q_{f_i}\right)} \otimes \text{SWAP}\left(j + q_i, j + n + q_{f_i}\right) \otimes I^{\otimes(j+q_i)}\right),$$

$$q_{f_i} = \sum_{k=0}^{i-1} n_{f_k}, q_i = \sum_{k=0}^{i-1} n_k, n_f = \sum_{k=0}^{d-1} n_{f_k}, \text{ and } n = \sum_{k=0}^{d-1} n_k$$
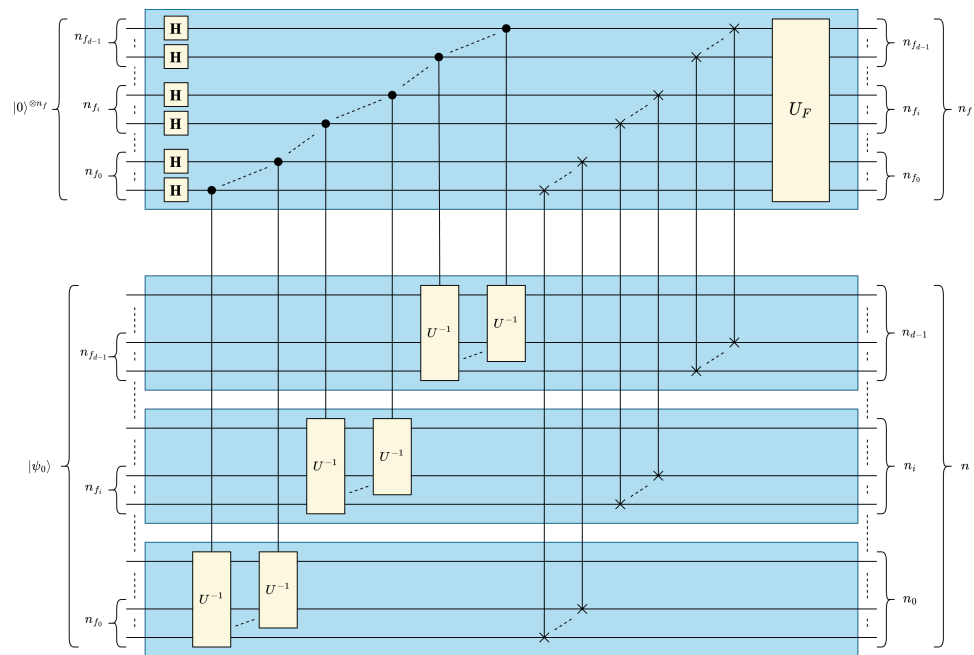
Figure 6. Generalized multidimensional quantum convolution circuit.

Circuit Depth Analysis of Generalized Multidimensional Quantum Convolution

As a result of the "*stacked*" structure, the data of all d dimensions could be concurrently processed in parallel. Consequently, the circuit depth of the multidimensional quantum circuit becomes dependent on the largest data dimension $N_{\text{max}} = 2^{n_{\text{max}}}$, where $n_{\text{max}} = \max_{i=0}^{d-1}(n_i)$, in lieu of the total data size N. The circuit component of the optimized 1-D circuit with the greatest depth contribution U_{mux} is performed in parallel on each dimension in the generic d-D circuit. Specifically, U_{mux} scales with the number of qubits used to represent the largest data dimension $n_{\text{max}} = \lceil \log_2 N_{\text{max}} \rceil$. Note that the

parallelization across dimensions applies to the Hadamard and SWAP operations from (5) and (9), see Figure 6, and therefore these operations are of constant depth complexity, i.e., $\mathcal{O}(1)$. The depth complexity of the multidimensional U_F operation is determined by the total number of elements in the filter N_F, and therefore the C2Q-based implementation of U_F does not benefit from multidimensional stacking. Accordingly, the circuit depth of the generalized multidimensional quantum convolution operation could be derived from (22) and expressed in (25), where $n_{f_{\max}} = \max_{i=0}^{d-1}(n_{f_i})$ is the number of qubits representing the maximum filter dimension $N_{f_{\max}} = 2^{n_{f_{\max}}}$. It is worth mentioning that the generic multidimensional formula in (25) reduces to the 1-D formula in (22) when $n = n_{\max}$ and $n_f = n_{f_{\max}}$:

$$
\mathcal{D}_{d\text{-D conv}}^{\text{opt}}(n, n_f) = \mathcal{O}\left(n_{f_{\max}} n_{\max}^2 - n_{f_{\max}}^2 n_{\max} + 2^{n_f}\right), \text{ where}
$$
$$
n_{\max} = \max_{i=0}^{d-1}(n_i), \ n_{f_{\max}} = \max_{i=0}^{d-1}(n_{f_i}), \text{ and } n_{\max} \gg n_{f_{\max}}
$$

(25)

4. Experimental Setup and Results

We experimentally demonstrate our proposed technique for generalized, multidimensional quantum convolution with unity stride on real-world, high-resolution, multidimensional image data, see Figure 7. By leveraging the `Qiskit` SDK (v0.39.4) from IBM Quantum [9], we simulate our quantum circuits in the following formats: (1) classically (to present the ideal/theoretical expectation), (2) noise-free (using statevector simulation), and (3) noisy (using 1,000,000 "shots" or circuit samples). Moreover, we present a quantitative comparison of the obtained results using fidelity [20] as a similarity metric between compared results ρ and σ, see (26). Experiments were performed on a 16-core AMD EPYC 7302P CPU with frequencies up to 3.3 GHz, 128 MB of L3 cache, and access to 256 GB of DDR4 RAM. In our analysis, we evaluated the correctness of the proposed techniques by comparing classical results with noise-free results. We also evaluated the scalability of the proposed techniques for higher-resolution images by comparing the classical results with both the noise-free and noisy results. Finally, we plotted the circuit depth of our techniques with respect to the data size and filter size as shown in Figure 8.

(**a**) Black-and-white (B/W) image.　　　(**b**) Color (RGB) Image

Figure 7. Real-world, high-resolution, multidimensional images used in experimental trials.
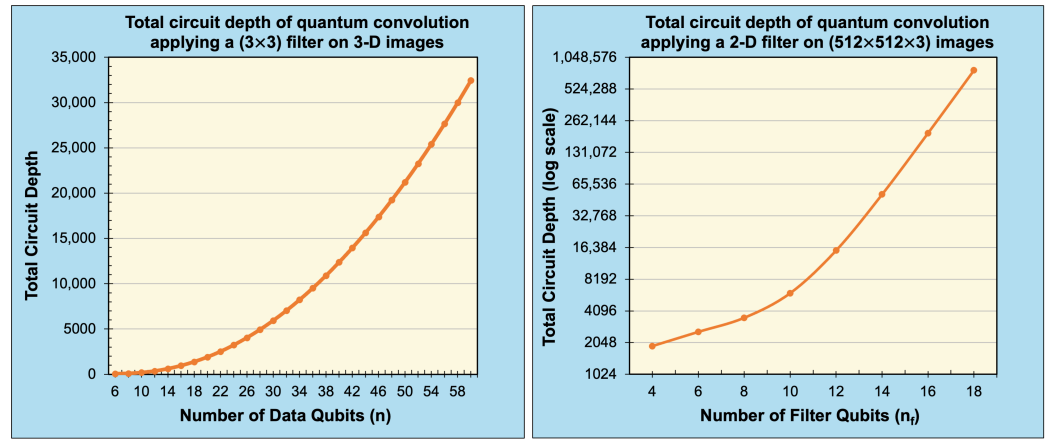
(**a**) Depth with respect to data size. (**b**) Depth with respect to filter size.

Figure 8. Circuit depth of quantum convolution with respect to data and filter qubits.

$$\text{Fidelity}(\rho, \sigma) = \text{tr} \sqrt{\sqrt{\rho} \cdot \sigma \cdot \sqrt{\rho}} \tag{26}$$

In our experiments, we evaluated our techniques using well-known (3×3) and (5×5) filters, i.e., Averaging F_{avg}, Sobel edge-detection $F_{\text{Sx}}/F_{\text{Sy}}$, Gaussian blur F_{blur}, and Laplacian of Gaussian blur (Laplacian) $F_{\mathcal{L}}$, see (27)–(30). We applied zero padding to maintain the size of the filter dimensions at a power of two for quantum implementation. In addition, we used wrapping to resolve the boundary conditions, and we restricted the magnitude of the output between $[0, 255]$ to mitigate quantization errors in the classical domain:

$$F_{\text{avg}}^{3 \times 3} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad F_{\text{avg}}^{5 \times 5} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{27}$$

$$F_{\text{blur}}^{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \qquad F_{\text{blur}}^{5 \times 5} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{28}$$

$$F_{\text{Sx}} = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad F_{\text{Sy}} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{29}$$

$$F_{\mathcal{L}}^{3 \times 3} = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad F_{\mathcal{L}}^{5 \times 5} = \frac{1}{20} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -24 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{30}$$

We applied 2-D convolution filters to black-and-white (B/W) and RGB images, see Figure 7, ranging in resolution from (8×8) pixels to (512×512) pixels and $(8 \times 8 \times 3)$ pixels to $(512 \times 512 \times 3)$ pixels, respectively. The number of filter qubits can be obtained by the size of filter dimensions, i.e., $n_f = \lceil \log_2 3 \rceil + \lceil \log_2 3 \rceil = 4$ qubits for (3×3) filters and $n_f = \lceil \log_2 5 \rceil + \lceil \log_2 5 \rceil = 6$ qubits for (5×5) filters. Therefore, our simulated quantum circuits ranged in size $(n + n_f)$ from 10 qubits to 26 qubits. Figures 9 and 10 present the reconstructed output images for classical, noise-free, and noisy experiments using (128×128) and $(128 \times 128 \times 3)$-pixel input images, respectively.

To demonstrate our method's applicability to data and filters of any dimensionality, we also performed experiments applying the 1-D and 3-D averaging filter to 1-D real-world audio data and 3-D real-world hyperspectral data, respectively. The audio files were sourced from the publicly available sound quality assessment material published by the European Broadcasting Union and modified to be single channel with data sizes ranging from 2^8 values to 2^{20} values when sampled at 44.1 kHz [21]. Figures 11 and 12 present the reconstructed output images and fidelity, respectively, from applying (3) and (5) averaging filters. The hyperspectral images were sourced from the Kennedy Space Center (KSC) dataset [22] and resized to range from $(8 \times 8 \times 8)$ pixels to $(128 \times 128 \times 128)$ pixels. Figures 13 and 14 present the reconstructed output images and fidelity, respectively, from applying $(3 \times 3 \times 3)$ and $(5 \times 5 \times 5)$ averaging filters.

Comparison of the noise-free quantum results against the ideal classical results demonstrates a 100% fidelity across all trials. Thus, in a noise-free (statevector) environment, our proposed quantum convolution technique correctly performs an identical operation to classical convolution given the same input parameters and boundary conditions.

When considering the behavior of noisy (sampled) environments, Figures 12, 14 and 15 plot the fidelity of the noisy quantum results against the ideal classical results. We observe a monotonic decrease in fidelity as the data size (image resolution) increases, consistent with previously reported behavior [23]. Such behavior derives from how the number of shots required to properly characterize a quantum state increases with the corresponding number of qubits in order to reduce the effects of statistical noise. Notably, the fidelity varies dramatically depending on the filter category and size, where the largest discrepancy occurs between the (5×5) Averaging and (5×5) Laplacian filters for a data size of 65,536 values. Specifically, for a B/W image of (256×256) pixels, the Averaging filter had a fidelity of 94.84%, while the Laplacian filter had a fidelity of 8.82%—a difference of 86.02%, see Figure 15a. In general, we observed that the average and blur filters perform better than the edge detection methods (Sobel/Laplacian). Since the output data are reconstructed from only a portion of the final state $|\psi_5\rangle$, it is likely that sparse filters, represented in Figures 9 and 10 as being mostly black, are significantly less likely to be recorded during sampled measurement, resulting in reduced fidelity. For practical applications, dimension reduction techniques, such as pooling, can be used to mitigate information loss [23].
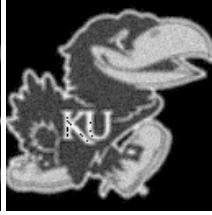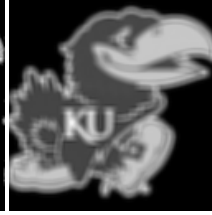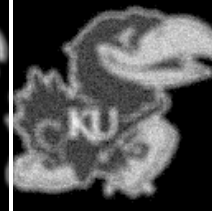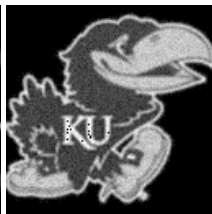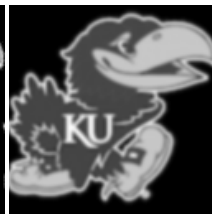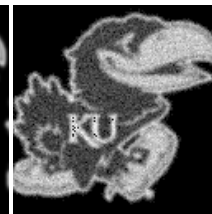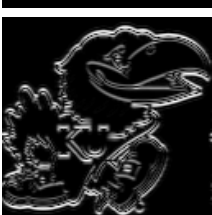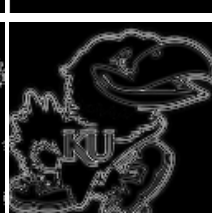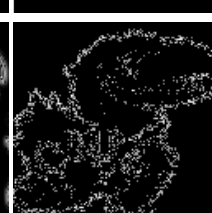
Filter	(3×3) filter Classical/ Noise-Free	(3×3) filter Noisy $(10^6$ shots$)$	(5×5) filter Classical/ Noise-Free	(5×5) filter Noisy $(10^6$ shots$)$
Average				
Gaussian				
Sobel-X			Not Applicable	Not Applicable
Sobel-Y			Not Applicable	Not Applicable
Laplacian				

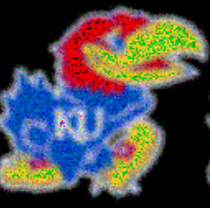**Figure 9.** The 2-D convolution filters applied to a (128×128) B/W image.
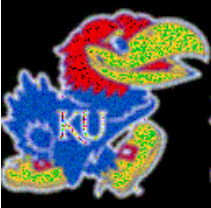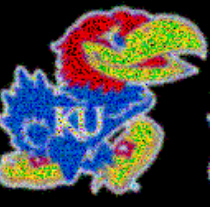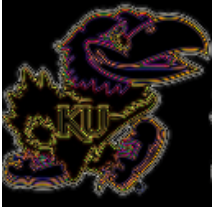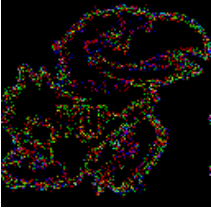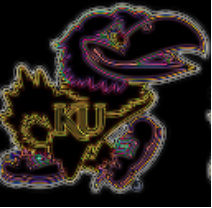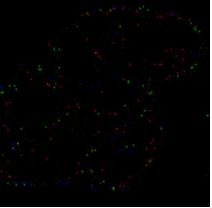
Filter	(3×3) filter Classical/ Noise-Free	(3×3) filter Noisy $(10^6$ shots$)$	(5×5) filter Classical/ Noise-Free	(5×5) filter Noisy $(10^6$ shots$)$
Average				
Gaussian				
Sobel-X			Not Applicable	Not Applicable
Sobel-Y			Not Applicable	Not Applicable
Laplacian				

Figure 10. The 2-D convolution filters applied to a $(128 \times 128 \times 3)$ RGB image.

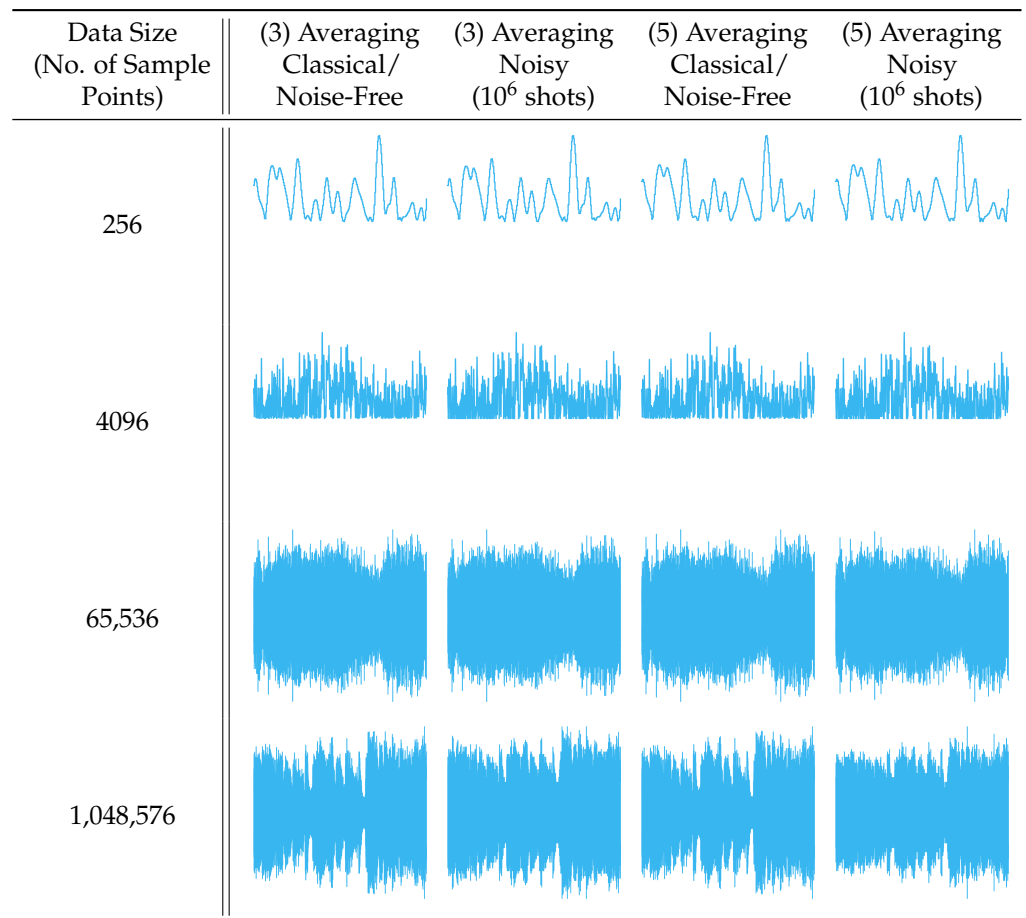| Data Size (No. of Sample Points)	(3) Averaging Classical/ Noise-Free	(3) Averaging Noisy (10^6 shots)	(5) Averaging Classical/ Noise-Free	(5) Averaging Noisy (10^6 shots)
256				
4096				
65,536				
1,048,576				

Figure 11. The 1-D convolution (averaging) filters applied to 1-D audio samples.

Figure 12. Fidelity of 1-D convolution (averaging) filters with unity stride on 1-D audio data (sampled with 1,000,000 shots).

Data Size ($N_{rows} \times N_{cols} \times N_{bands}$)	$(3 \times 3 \times 3)$ Averaging Classical/ Noise-Free	$(3 \times 3 \times 3)$ Averaging Noisy (10^6 shots)	$(5 \times 5 \times 5)$ Averaging Classical/ Noise-Free	$(5 \times 5 \times 5)$ Averaging Noisy (10^6 shots)
$(8 \times 8 \times 8)$				
$(16 \times 16 \times 16)$				
$(32 \times 32 \times 32)$				
$(64 \times 64 \times 64)$				
$(128 \times 128 \times 128)$				

Figure 13. The 3-D convolution (averaging) filters applied to 3-D hyperspectral images (bands 0, 1, and 2).

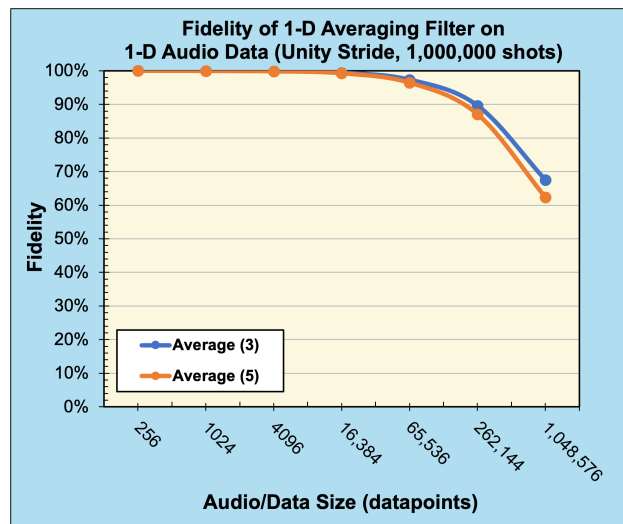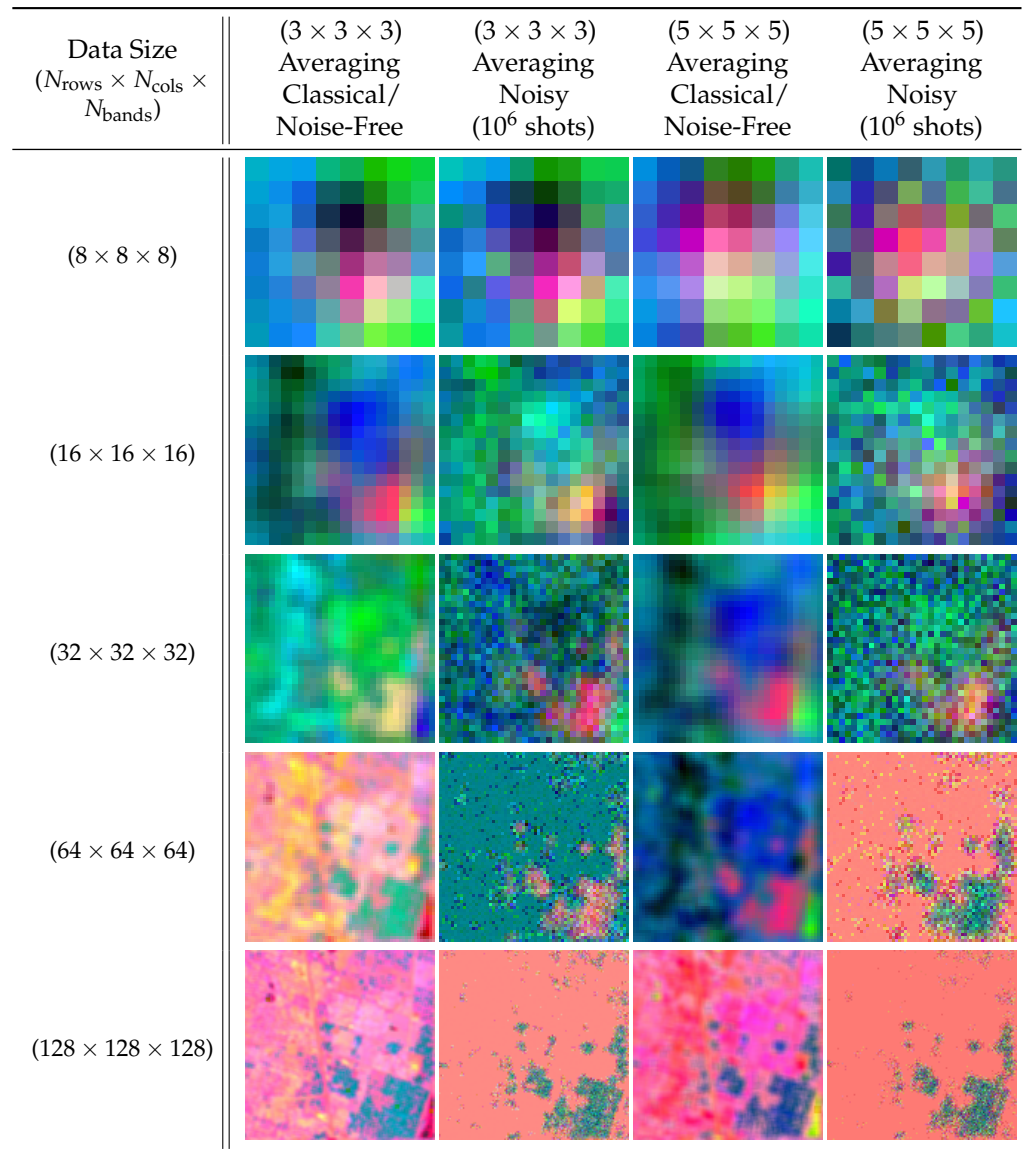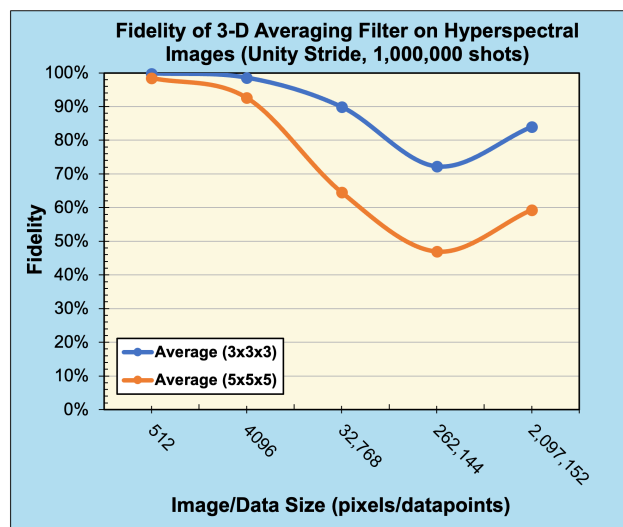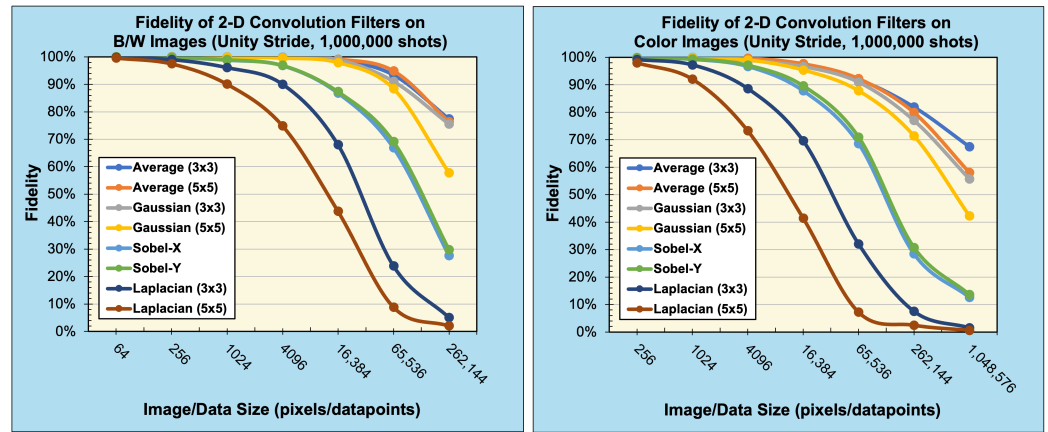**Figure 14.** Fidelity of 3-D convolution (averaging) filters with unity stride on 3-D hyperspectral data (sampled with 1,000,000 shots).

(**a**) Black-and-white (B/W) images	(**b**) Color (RGB) images

Figure 15. Fidelity of 2-D convolution with unity stride (sampled with 1,000,000 shots).

5. Discussion

In the following section, we compare our proposed method of quantum convolution to the related work discussed in Section 2.1 in terms of filter generalization and circuit depth.

5.1. Arbitrary Multidimensional Filtering

Our generalizable and parameterized technique of quantum convolution with unity stride offers distinct workflow advantages over existing *fixed-filter* quantum convolution techniques in variational applications, such as quantum machine learning. Our technique does not require extensive development for each new filter design. For instance, current quantum convolutional filters are primarily two dimensional, only focusing on image processing. However, the development of even similar filters targeting audio and video processing, for example, would require extensive development and redesign. Our method offers a systematic and straightforward approach for generating practical quantum circuits given fundamental input variables.

5.2. Circuit Depth

Figure 8a,b show the circuit depth for our proposed technique of generalized quantum convolution with respect to the total number of data qubits $n = \sum_{i=0}^{d-1}(n_i)$ and the total number of filter qubits $n_f = \sum_{i=0}^{d-1}(n_{f_i})$, respectively. The results were gathered using the `QuantumCircuit.depth()` method built into `Qiskit` for a `QuantumCircuit` transpiled to fundamental single-qubit and CNOT quantum gates. Figure 8a illustrates quadratic circuit depth complexity with respect to the data qubits n for a fixed filter size $N_F = 2^{n_f}$, aligning with our theoretical expectation in (25). Note that $n = \sum_{i=0}^{d-1} n_i$ and $n_{\max} = \max_{i=0}^{d-1}(n_i)$ for d-dimensional data. Similarly, Figure 8b (plotted on a log-scale) illustrates exponential circuit depth complexity with respect to n_f for a fixed data size $N = 2^n$, which also aligns with our theoretical expectation in (25).

The time complexity comparison of our proposed quantum convolution technique against related work is shown in Table 1. Compared to classical direct implementations on CPUs, our proposed technique for generalized quantum convolution demonstrates an exponential improvement with respect to data size $N = 2^n$, i.e., $\mathcal{O}(n^2)$ vs. $\mathcal{O}(N^2)$, see (25) and Table 1c. As discussed in Section 2.1, the fastest classical GEMM implementation of convolution on GPUs (excluding data I/O overhead) [12,13] has a complexity of $\mathcal{O}(N_F N)$, see Table 1c. Even when including quantum data encoding, which is equivalent to data I/O overhead, our method remains to demonstrate a linear improvement with respect to data size N by a factor of the filter size N_F, see (31), over classical GEMM GPUs:

$$\begin{aligned}
\mathcal{D}_{\text{proposed}}(n) &= \mathcal{D}_{\text{C2Q}}(n) + \mathcal{D}_{d\text{-D conv}}^{\text{opt}}(n) \\
&= \mathcal{O}\left(2^n + n_{\max}^2\right) = \mathcal{O}(2^n) = \mathcal{O}(N), \text{ for fixed } n_f
\end{aligned} \tag{31}$$

Compared to *fixed-filter* quantum convolution techniques [4–8], our proposed arbitrary filter quantum convolution technique (for unity stride) demonstrates improved circuit depth complexity with respect to data size when factoring in the circuit depth contribution from data encoding. For a fixed filter, i.e., n_f is constant, the depth of the proposed method scales quadratically with the largest data dimension $n_{\max}$, see (25) and (31). As described in Section 2.1, *fixed-filter* quantum convolution techniques similarly show quadratic depth scaling with respect to the number of qubits n, see Table 1b. For data encoding, the *fixed-filter* techniques use either the FRQI [14] or NEQR [15] algorithms, which have circuit depth complexities of $\mathcal{O}(4^n)$ or $\mathcal{O}(qn2^n)$, respectively. In contrast, our proposed technique uses C2Q data encoding [17], which has a depth complexity of $\mathcal{O}(2^n)$—a quadratic and linear improvement over FRQI and NEQR, respectively, see Table 1a.

Table 1. Comparison of depth/time complexity of proposed generalized quantum convolution technique against related work.

a Depth complexity of quantum data encoding (I/O) techniques		
FRQI [14]	NEQR [15]	C2Q [17]
$\mathcal{O}_{\text{I/O}}(4^n)$	$\mathcal{O}_{\text{I/O}}(qn2^n)$	$\mathcal{O}_{\text{I/O}}(2^n)$
b Depth complexity of quantum convolution algorithms for a fixed filter		
Proposed	Related Work [4–8]	
$\mathcal{O}_{\text{alg}}(n_{\max}^2)$	$\mathcal{O}_{\text{alg}}(n^2)$	
c Complexity of proposed technique compared to classical convolution		
Proposed	Direct (CPU) [10]	FFT (CPU/GPU) [10,11]	GEMM (GPU) [12,13]
$\mathcal{O}_{\text{alg}}\left(\begin{array}{c} n_{f_{\max}} n_{\max}^2 - n_{f_{\max}}^2 n_{\max} \\ +2^{n_f} \end{array}\right)$ $\mathcal{O}_{\text{alg + I/O}}\left(\begin{array}{c} n_{f_{\max}} n_{\max}^2 - n_{f_{\max}}^2 n_{\max} \\ +2^n + 2^{n_f} \end{array}\right)$	$\mathcal{O}_{\text{alg}}(N^2) \equiv$ $\mathcal{O}_{\text{alg}}(4^n)$	$\mathcal{O}_{\text{alg}}(N \log N) \equiv$ $\mathcal{O}_{\text{alg}}(n2^n)$	$\mathcal{O}_{\text{alg}}(N_F N) \equiv$ $\mathcal{O}_{\text{alg}}\left(2^{(n+n_f)}\right)$

6. Conclusions

In this work, we proposed and evaluated a method for generalizing the convolution operation with arbitrary filtering and unity stride for multidimensional data in the quantum domain. We presented the corresponding circuits and their performance analyses along with experimental results that were collected using the IBM `Qiskit` development environment. In our experimental work, we validated the correctness of our method by comparing classical results to noise-free quantum results. We also demonstrated the practicality of our method for various convolution filters by evaluating the noisy quantum results. Furthermore, we presented experimentally verified analyses that highlight our technique's advantages in terms of time complexity and/or circuit depth complexity compared to existing classical and quantum methods, respectively. Future work will focus on adapting our proposed technique for arbitrary strides. In addition, we will investigate multidimensional quantum machine learning as a potential application of our proposed technique.

Author Contributions: Conceptualization: M.J., V.J., D.K. and E.E.-A.; Methodology: M.J., V.J., D.K. and E.E.-A.; Software: M.J., D.L., D.K. and E.E.-A.; Validation: M.J., V.J., D.L., D.K. and E.E.-A.; Formal analysis: M.J., V.J., D.L., D.K. and E.E.-A.; Investigation: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Resources: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Data curation: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Writing—original draft preparation: M.J., A.N., V.J., D.L., D.K., M.C., I.I. and E.E.-A.; Writing—review and editing: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Visualization: M.J., A.N., V.J., D.L., D.K., M.C.,

Abbreviations

The following abbreviations are used in this manuscript:

1-D	One Dimensional
C2Q	Classical to Quantum
FFT	Fast Fourier Transform
FRQI	Flexible Representation of Quantum Images
GEMM	General Matrix Multiplication
NEQR	Novel Enhanced Quantum Representation
QHED	Quantum Hadamard Edge Detection
QWT	Quantum Wavelet Transform

References

1. Rhodes, W. Acousto-optic signal processing: Convolution and correlation. *Proc. IEEE* **1981**, *69*, 65–79. [CrossRef]
2. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [CrossRef]
3. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256. [CrossRef]
4. Fan, P.; Zhou, R.G.; Hu, W.; Jing, N. Quantum image edge extraction based on classical Sobel operator for NEQR. *Quantum Inf. Process.* **2018**, *18*, 24. [CrossRef]
5. Ma, Y.; Ma, H.; Chu, P. Demonstration of Quantum Image Edge Extration Enhancement Through Improved Sobel Operator. *IEEE Access* **2020**, *8*, 210277–210285. [CrossRef]
6. Zhang, Y.; Lu, K.; Gao, Y. QSobel: A novel quantum image edge extraction algorithm. *Sci. China Inf. Sci.* **2015**, *58*, 1–13. [CrossRef]
7. Zhou, R.G.; Yu, H.; Cheng, Y.; Li, F.X. Quantum image edge extraction based on improved Prewitt operator. *Quantum Inf. Process.* **2019**, *18*, 261. [CrossRef]
8. Li, P. Quantum implementation of the classical Canny edge detector. *Multimed. Tools Appl.* **2022**, *81*, 11665–11694. [CrossRef]
9. IBM Quantum. Qiskit: An Open-source Framework for Quantum Computing. 2021. Available online: https://zenodo.org/records/7416349 (accessed on 19 October 2023).
10. Burrus, C.S.; Parks, T.W. *DFT/FFT and Convolution Algorithms: Theory and Implementation*, 1st ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1991.
11. Podlozhnyuk, V. FFT-based 2D convolution. *NVIDIA* **2007**, *32*. Available online: https://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_64_website/projects/convolutionFFT2D/doc/convolutionFFT2D.pdf (accessed on 19 October 2023).
12. NVIDIA. Convolution Algorithms. 2023. Available online: https://docs.nvidia.com/deeplearning/performance/dl-performance-convolutional/index.html#conv-algo (accessed on 19 October 2023).
13. NVIDIA. CUTLASS Convolution. 2023. Available online: https://github.com/NVIDIA/cutlass/blob/main/media/docs/implicit_gemm_convolution.md (accessed on 19 October 2023).
14. Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **2011**, *10*, 63–84. [CrossRef]
15. Zhang, Y.; Lu, K.; Gao, Y.; Wang, M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **2013**, *12*, 2833–2860. [CrossRef]
16. Yao, X.W.; Wang, H.; Liao, Z.; Chen, M.C.; Pan, J.; Li, J.; Zhang, K.; Lin, X.; Wang, Z.; Luo, Z.; et al. Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment. *Phys. Rev. X* **2017**, *7*, 031041. [CrossRef]

17. El-Araby, E.; Mahmud, N.; Jeng, M.J.; MacGillivray, A.; Chaudhary, M.; Nobel, M.A.I.; Islam, S.I.U.; Levy, D.; Kneidel, D.; Watson, M.R.; et al. Towards Complete and Scalable Emulation of Quantum Algorithms on High-Performance Reconfigurable Computers. *IEEE Trans. Comput.* **2023**, *72*, 2350–2364. [CrossRef]
18. Li, X.; Yang, G.; Torres, C.M.; Zheng, D.; Wang, K.L. A Class of Efficient Quantum Incrementer Gates for Quantum Circuit Synthesis. *Int. J. Mod. Phys. B* **2014**, *28*, 1350191. [CrossRef]
19. Balauca, S.; Arusoaie, A. Efficient Constructions for Simulating Multi Controlled Quantum Gates. In Proceedings of the Computational Science—ICCS 2022, London, UK, 21–23 June 2022; Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot, P.M.A., Eds.; Springer: Cham, Switzerland, 2022; pp. 179–194.
20. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK, 2010; p. 409. [CrossRef]
21. Geneva, S. Sound Quality Assessment Material: Recordings for Subjective Tests. 1988. Available online: https://tech.ebu.ch/publications/sqamcd (accessed on 19 October 2023).
22. Graña, M.; Veganzons, M.A.; Ayerdi, B. Hyperspectral Remote Sensing Scenes. Available online: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_(KSC) (accessed on 19 October 2023).
23. Jeng, M.; Islam, S.I.U.; Levy, D.; Riachi, A.; Chaudhary, M.; Nobel, M.A.I.; Kneidel, D.; Jha, V.; Bauer, J.; Maurya, A.; et al. Improving quantum-to-classical data decoding using optimized quantum wavelet transform. *J. Supercomput.* **2023**, *79*, 20532–20561. [CrossRef]