Quantum Edge Detection

Application of a QED Method using Qiskit

Abstract

Quantum Image Processing (QImP) is an emerging field of quantum computing that can strengthen capacity for storing, processing, and retrieving information from images and video. While not as popular as other surging quantum topics (encryption, optimization, quantum internet), QImP has the potential to advance emerging deep tech fields like autonomous vehicles, medical imaging, and security applications such as facial recognition, gait analysis, "DeepFake" video and image manipulation detection. Existing research shows QImP offers significant advantages over current classical methods of image processing by leveraging the unique computational power of quantum computers to make processing tasks faster and more efficient. This effort will utilize existing quantum image processing algorithms combined with classical pre- and post-processing techniques to determine if a processing pipeline can be developed from beginning to end. The execution of quantum circuits were conducted against quantum simulators

Introduction: Objective

Can we begin to leverage today's quantum computers in image processing tasks? Edge detection is a basic image processing task that can provide highly useful information for machine learning and artificial intelligence tasks. This effort takes a basic quantum edge detection algorithm presented in prior research and extends the published sample to larger image sizes. Further, this effort completes an exercise to handle these larger images using a pre-processing technique whereby images are broken down into smaller images, processed using an quantum edge detection algorithm and then re-combined to present the edge-detected image in its original size.

Previous Research

Edge detection is primary image processing function that is utilized by many systems in order to feed into image recognition and image classification systems. It is therefore worthwhile to attempt to further improve the speed at which edge-detection can occur in images. For this study the same image representation method used in the referenced study. That image representation is called Quantum Probability Image Encoding or QPIE. In addition, the method for edge detection used is the QHED (Quantum Hadamard Edge Detection) algorithm. This work expands on the previous research by allowing for larger images.

Implementation

The implementation consisted of the creation of a single Python application in the form of a Jupyter notebook. The Python application utilizes the IBM Qiskit quantum programming library. The Qiskit library was used to execute the QHED quantum edge detection algorithm against IBM quantum simulators. The python programs presented in the original work were heavily modified to accept images of larger sizes and the portions dealing with image chunking and re-combining are unique to this body of work. In addition, the computation of required qubits was introduced based on the size of the target image.

Amplitude Encoding

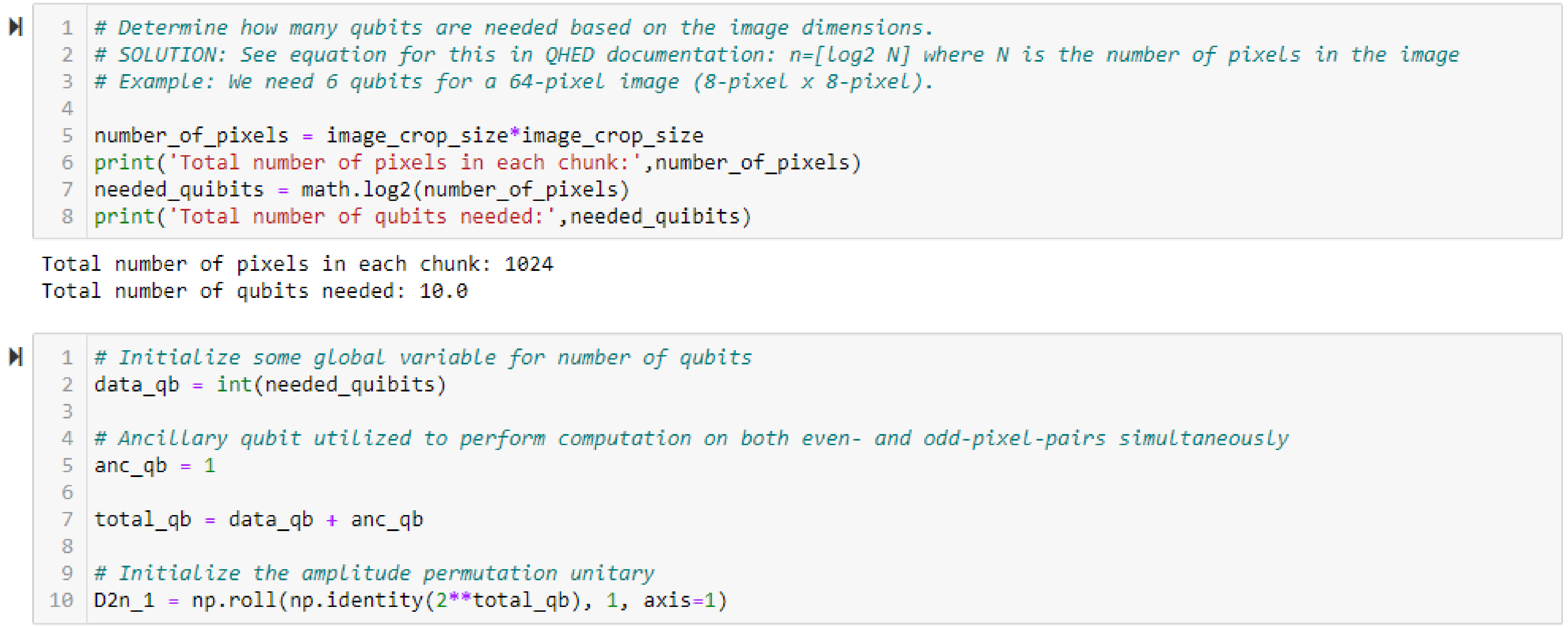
There are several methods for encoding an image into a quantum state. The most common are Angle encoding, Basis encoding, and Amplitude encodings. This work utilizes Amplitude encoding whereby the amplitudes of a quantum system represent our data values. Every quantum system be described by a wavefunction and that wavefunction is what is used to encode the data.

Pre- and Post-Processing

The implementation consisted of the creation of a single Python application in the form of a Jupyter notebook. The Python application utilizes the IBM Qiskit quantum programming library. The Qiskit library was used to execute the QHED quantum edge detection algorithm against IBM quantum simulators, IBM “fake” quantum computers, which closely mimic a real quantum computer and finally against real IBM quantum computes with various numbers of qubits. In order to process images of 255 x 255 pixels, images had to be broken down into smaller images and the QHED algorithm was applied to each sub-image. In post-processing, the sub-images were recombined into an image of the original size.

Pre- and Post-Processing

Qubit Prep Work



```python
# Determine how many qubits are needed based on the image dimensions.
# SOLUTION: See equation for this in QHED documentation: n=[log2 N] where N is the number of pixels in the image
# Example: We need 6 qubits for a 64-pixel image (8-pixel x 8-pixel).

number_of_pixels = image_crop_size*image_crop_size
print('Total number of pixels in each chunk:',number_of_pixels)
needed_quibits = math.log2(number_of_pixels)
print('Total number of qubits needed:',needed_quibits)
```

```
Total number of pixels in each chunk: 1024
Total number of qubits needed: 10.0
```

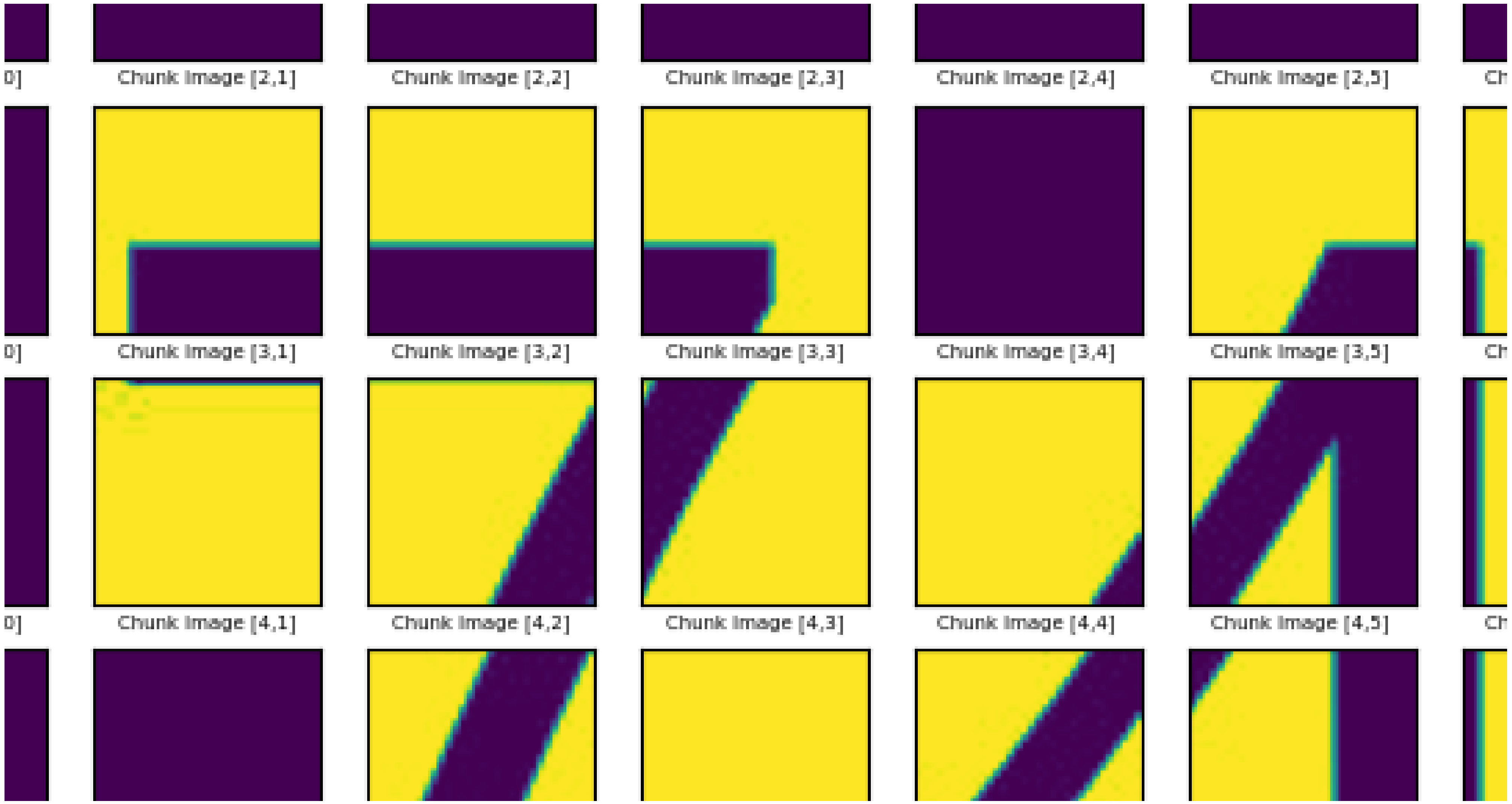
```python
# Initialize some global variable for number of qubits
data_qb = int(needed_quibits)

# Ancillary qubit utilized to perform computation on both even- and odd-pixel-pairs simultaneously
anc_qb = 1

total_qb = data_qb + anc_qb

# Initialize the amplitude permutation unitary
D2n_1 = np.roll(np.identity(2**total_qb), 1, axis=1)
```

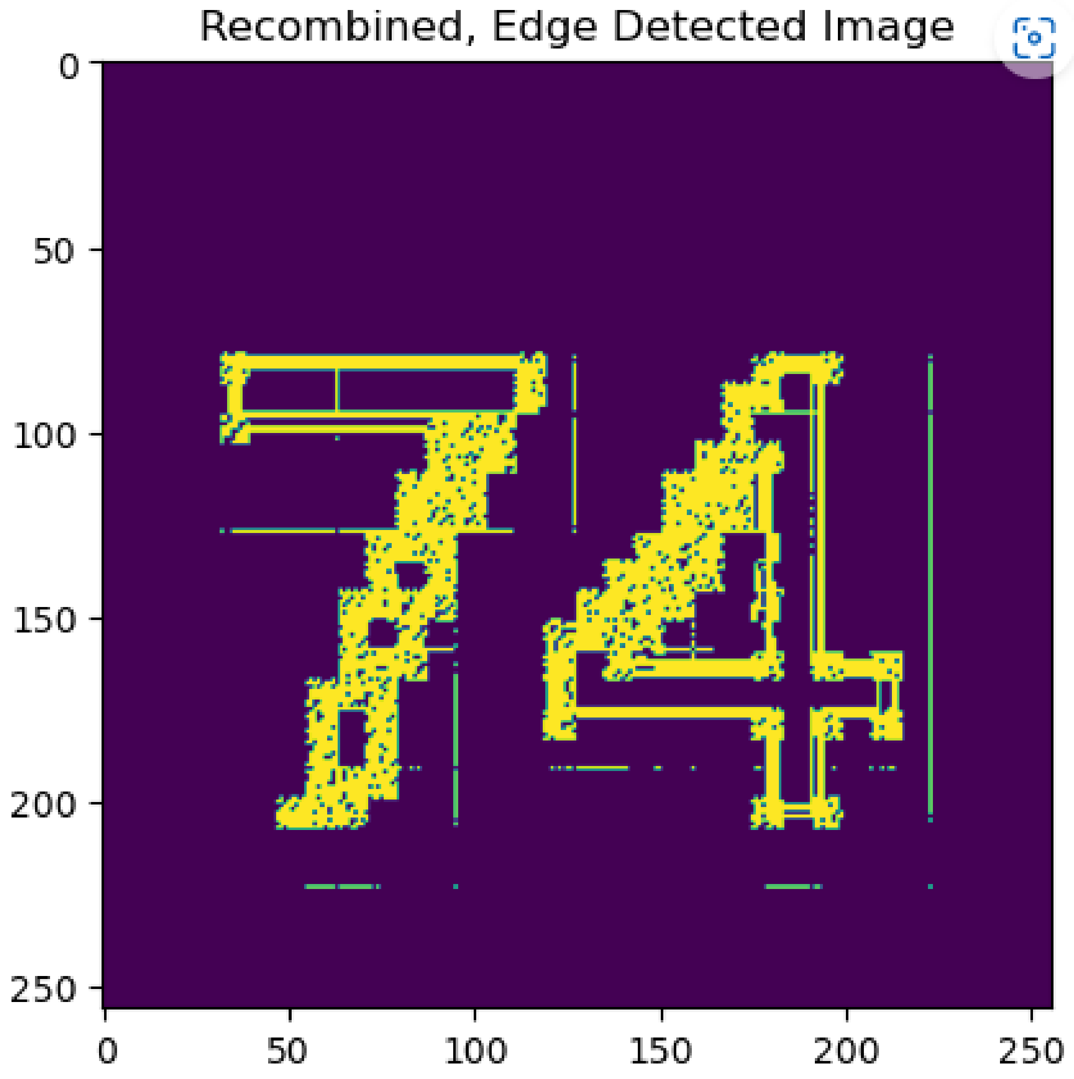
Pre- and Post-Processing



Results

The results when our circuit was executed against a simulator were as expected although not perfect. Some further adjustments to the algorithm would need to be performed in order to provide better results. Experiments were not run against a actual quantum computer at this time as a majority of the available quantum systems from IBM that are free consist of less than 10 qubits and these experiments require at least 11. Of the IBM quantum systems that do have higher numbers of qubits, many are restricted and if available have hundreds to thousands of jobs waiting in queue therefore are deemed unavailable for the purposes of this exercise.

Results



Recombined, Edge Detected Image
0
50
100
150
200
250
0
50
100
150
200
250

Conclusions

Current implementations of quantum image processing algorithms do not provide usable results at this time, but advancements are being made every day. Many of the advancements come in the form of **error correction**. In addition, new types of quantum computers are being created with better **gate fidelity** and **coherence times** which will improve results.