Quantum Image Processing and Compression

Frederic Wang

May 6, 2022

1 Overview of Quantum Image Processing

Quantum image processing strives to use quantum circuits to analyze, represent, and compress digital images with better time and space efficiency than classical approaches. The idea behind quantum image processing was first brought to light in 2003 by Beach et al. [1], where they considered using Grover's algorithm (in theory) to achieve compression beyond classical approaches. In 2017, Yao et. al [2] created an efficient algorithm for image edge detection through the use of quantum circuits, with real experimental tests backing up the algorithm.

With the recent advances in quantum image processing, it will certainly outperform classical image processing for specific tasks once quantum computers are available. In this paper, I will highlight different problems in which quantum methods do better and do worse than classical methods.

2 Quantum Image Processing Techniques

2.1 Image Encoding

In classical computing, images are represented through matrices. For instance, for a RGB image of height n pixels and width m pixels, we can create three matrices of size nxm where each element of the matrix represents the intensity of red, green, and blue color, respectively, of the corresponding pixel in the image. For simplicity, I decided to use 8-bit grayscale images for this project, where each image of height n and width m can be represented by a nxm matrix A. Entry $A_{i,j}$ of the matrix corresponds to pixel (i, j) and takes on an integer value between 0 and 255 (hence the 8-bit representation), where 0 represents pure black and 255 represents pure white.

In order to convert this classical representation into a qubit state representation, I used the technique presented by Yao et al. [2]. For an image of size 2^a x 2^b, we can first flatten the classical matrix representation into a column vector v of length 2^{a+b}, and normalize it such that $||v|| = 1$. Then, we can prepare a superposition of $a + b$ qubits such that state $|k\rangle$ has coefficient v_k, so our prepared qubit state becomes $\sum_{i=0}^{2^{a+b}} v_i |i\rangle$. According to Soklakov and Schack [3], such a state can be prepared with poly$(a + b)$ time, which scales exponentially lower than the number of pixels of the image. In addition, the image itself now takes $a + b$ qubits to store rather than 2^{a+b} classical bits, an exponential improvement.

2.2 Quantifying Image Similarity

Image similarity can be implemented efficiently through quantum circuits. Using the encoding described above, if we want to compare two images that are represented with n qubits each, we can construct a image similarity circuit with $2n + 1$ qubits, where we have one ancilla bit and $2n$ qubits corresponding to the two images. Using the ancilla bit, we can simply perform a SWAP test between the two image states, so upon measuring the ancilla bit afterwards we get a probability of measuring 1 as $\frac{1}{2} - \frac{1}{2}| \langle IMAGE1|IMAGE2\rangle |^2$. The image similarity circuit can be seen in Figure 1.

Thus, two identical images, whose qubit state representations would be also identical, would give us a probability of $\frac{1}{2} - \frac{1}{2}(1) = 0$ of measuring a 1. On the other hand, two completely orthogonal images, which would have an inner product of approximately 0, would give us a probability of

$\frac{1}{2} - \frac{1}{2}(0) = \frac{1}{2}$ of measuring a 1. Of course, we would have to prepare and simulate the circuit multiple times in order to generate an accurate sample mean of the probability of measuring a 1.

To measure inner product similarity for a n-qubit or 2^n pixel image classically, we would have to compute 2^n multiplications. However, since each circuit run is independent, the variance of the sample mean of these measurements scales with $\frac{1}{measurements}$. Thus, we can calculate the same metric with a polynomial error bound in polynomial time in the quantum regime, an exponential speedup compared to the classical regime.

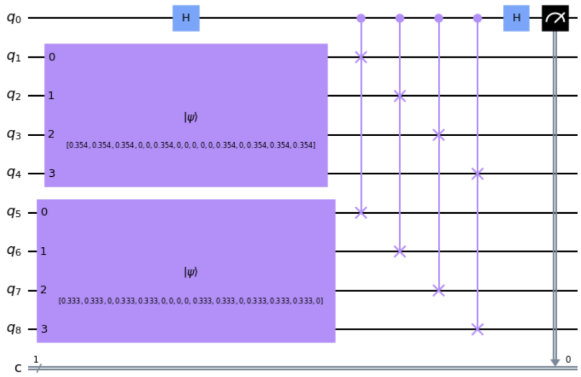


Figure 1: Image similarity circuit for two random 4-qubit images

For the next few experiments, 100 different circuit simulations were performed for each image to try and estimate the true probability of measuring a 1. When running the image similarity circuit on 100 pairs of identical images of size 4x4, we measure 0 all 100 times on average, as expected. On the other hand, running the circuit on 100 pairs of randomly generated images of size 4x4 (where each pixel has a $\frac{1}{2}$ chance of being black or white) resulted in measuring a 1 36.73 times on average, with a standard deviation of 8.91, meaning that on average two randomly generated images had an inner product of magnitude $|\langle IMG1|IMG2\rangle| = 0.51$, which makes sense as it is roughly in the middle of purely orthogonal images (inner product magnitude 0) and identical images (inner product magnitude 1)

2.3 Quantifying Rotational Symmetry

In order to calculate rotational symmetry, we can use a slightly modified version of the previous circuit. We input the same image as both arguments of the previous circuit, but then apply an X gate on each of the qubits for one of the images. This essentially changes our state from $\sum_{i=0}^{2^{a+b}} v_i |i\rangle$ to $\sum_{i=0}^{2^{a+b}} v_{2^{a+b}-i} |i\rangle$, which is the same as rotating the image by 180 degrees about the origin. The circuit can be seen in Figure 2. Similar modifications can be made for arbitrary rotations about the origin. From here, we can perform the same SWAP test mentioned above on the original image and the rotated image to measure the similarity of the image under rotation. Thus, we can also achieve an exponential speedup for calculating rotational symmetry with quantum circuits.

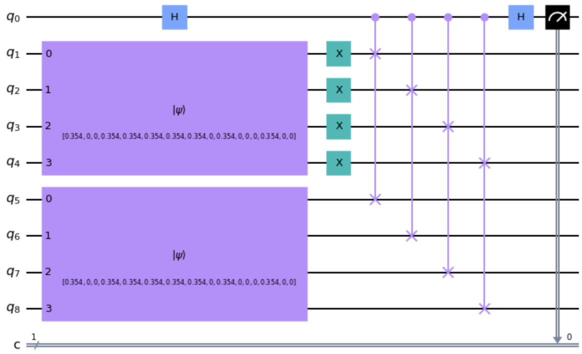


Figure 2: Symmetry detection circuit for a random 4-qubit image

Just like before, 100 circuit simulations were performed for each image to try and estimate the true probability of measuring a 1. When running the symmetry test on 100 fully symmetric 4x4 images, 0 is measured all 100 times, as expected. When running the symmetry test on 100 random 4x4 images, we measure 1 36.93 times on average with standard deviation 8.57, which is approximately the same as the image similarity results on random images from above.

3 Quantum Image Transforms

In classical image processing, there exist different image transforms to compress and sparsify image data. In this section, we highlight the quantum versions of the Fourier and Hadamard transforms.

3.1 Fourier Transform

In classical image processing, we can use the Discrete Fourier Transform to convert an image into its Fourier basis representation, which consists of mutually orthogonal sinusoids at equally spaced frequencies. In order to extract frequency coefficients of a signal a of length N, we can project our signal onto each basis vector:

$$A_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} a_n e^{-i2\pi\frac{kn}{N}}$$

In general, the lower frequency components of an image dominate the higher frequency components, so efficient classical methods for sparsifying and storing images in the frequency domain exist. While not exactly the same, this is the motivation behind compression techniques like JPEG [4]. In order to reconstruct the image from its frequency coefficients, we can apply the inverse Discrete Fourier Transform which multiplies each basis vector by its corresponding intensity:

$$a_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} A_k e^{i2\pi\frac{kn}{N}}$$

.

Similarly, there exists a quantum Fourier transform that takes as input a n-qubit state $|x\rangle = \sum_{i=0}^{2^n-1} x_i |i\rangle$ and outputs the frequency coefficients $|y\rangle \sum_{i=0}^{2^n-1} y_i |i\rangle$ defined as follows:

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{i2\pi nk/N}$$

To get the original state back, we can apply the inverse quantum Fourier transform:

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k e^{-i2\pi nk/N}$$

The best known algorithm for performing the classical Fourier Transform is the FFT (Fast Fourier Transform), which exploits the symmetry of the transform with a divide-and-conquer approach. For a classical image consisting of n pixels, we would need $O(nlog(n))$ time to compute the Fourier transform.

Since the Fourier transform is linear, we can represent it as a matrix operator, and decompose it into elementary quantum gates. The quantum circuit implementing the QFT is shown in Figure 3. Using the encoding method described in section 2.1, we can represent a n-pixel image with $\log_2(n)$ qubits. Since preparing the state takes $poly(\log_2(n))$ time, and we need $O(\log_2(n)^2)$ gates to perform the QFT, in theory we can perform a single QFT in $poly(\log_2(n))$ time.

However, if we wanted to accurately sample the distribution of the frequency coefficients or reconstructed image, we would have to run the circuit many times in order to get a good sample distribution of the qubit state. Also, because QFT frequency coefficients are generally complex, it is impossible to directly measure them, since the probability of measuring some state would be the squared amplitude of said complex coefficient. Thus, any sort of image processing with the quantum Fourier transform, or most quantum image transforms in general, must be done with the original qubits. I decided to perform some experiments seeing how many times we would have to run and measure the circuit for accurate image reconstruction in section 4.1.

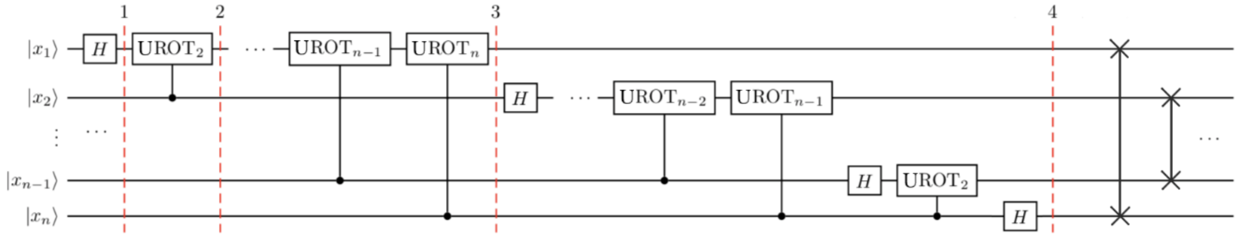


Figure 3: Quantum fourier transform circuit for n qubits [5]

3.2 Hadamard Transform

In classical signal processing, the Hadamard transform is defined recursively. The 1x1 Hadamard operator is defined as $H_0 = 1$, and for any $m > 0$ we define the $2^m x 2^m$ Hadamard operator as $H_m = \frac{1}{\sqrt{2}}\begin{bmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{bmatrix}$. Given an input vector a, the Hadamard transform of the vector is $H_k a$, where k is the next highest power of two after the length of a. (Usually, a would be padded with zeros so its length becomes the next highest power of two). Since the Hadamard operator is unitary, its inverse transform is the same as itself. In a similar manner to the Fourier transforms, we can calculate Hadamard coefficients of a length n image vector in time $O(nlog(n))$.

The quantum Hadamard (and inverse quantum Hadamard) transform can be implemented through the tensor product of n Hadamard gates, as shown in Figure 4. Thus, for a n pixel image vector, we only need $O(log(n))$ gates to implement both directions of the quantum Hadamard transform.

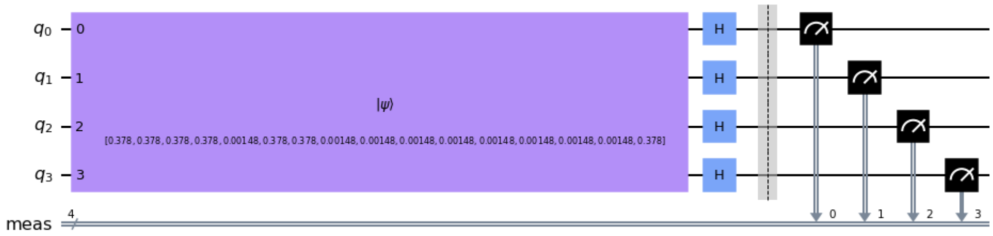


Figure 4: Hadamard transform circuit for a 4-qubit image

4 Quantum Image Reconstruction and Compression

In this section, we discuss the feasibility, efficiency and accuracy of using the quantum transforms introduced in the previous section to store and compress images.

4.1 Feasibility of Image Reconstruction

While encoding an image is relatively straightforward using the method from 2.1, decoding it from some output qubit state is not so easy, since we can only sample one value from the qubit state for each time the circuit is prepared and run. In this section, I will explore how many circuit simulations, with respect to size of input state, is necessary to generate an accurate reconstructed image without any intermediate processing steps, i.e. a QFT and IQFT is performed on the state like in Figure 5. (Of course, in a Qiskit simulation the input and output state of this example would be exactly the same, but in reality there would likely be some gate errors, so the results of these experiments are idealized.)

I decided to use RMSE (root mean square error) between the reconstructed image and input image, both of which are scaled accordingly so the maximum value in both is 255. I ran separate trials for 2x2, 4x4, 8x8, 16x16 images, and tried 100, 1000, 10000, 100000 circuit simulations and measurements for each size. For each experiment, I generated 100 random images like before, and

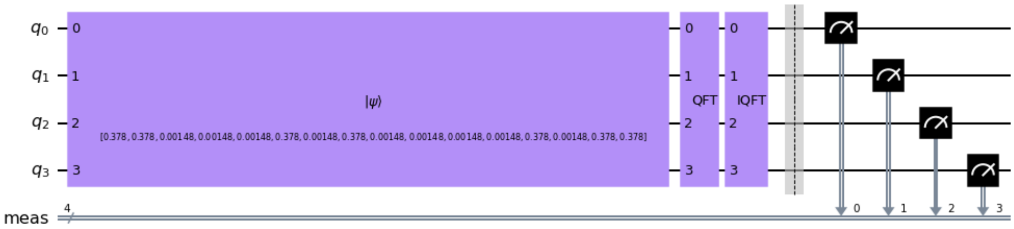
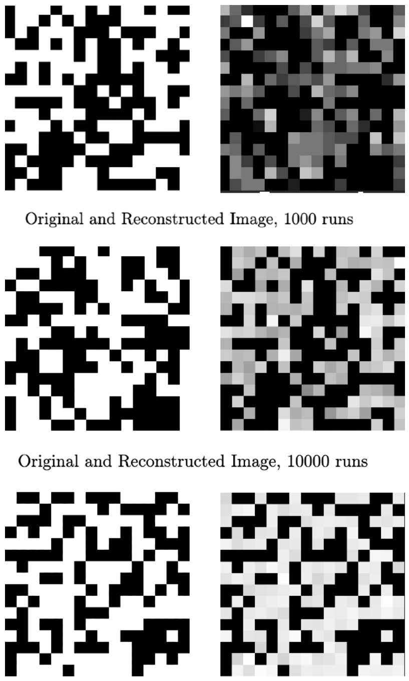


Figure 5: Circuit used in section 4.1


Original and Reconstructed Image, 1000 runs

Original and Reconstructed Image, 10000 runs

Original and Reconstructed Image, 100000 runs

Figure 6: Reconstructed 16x16 images for 1000, 10000, 100000 circuit simulations

I reported the average error metric over all 100 images. The numerical results can be seen in Table 1 and visual images can be seen in Figure 6.

Size of Image vs. # Circuit Runs	2x2 Image	4x4 Image	8x8 Image	16x16 Image
100 circuit runs	21.34±21.57	60.52±21.10	109.32±16.78	148.7±7.80
1000 circuit runs	9.09±8.29	23.29±10.14	54.78±11.08	96.79±8.33
10000 circuit runs	2.94±2.72	8.58±3.59	20.35±3.82	45.20±6.22
100000 circuit runs	1.11±0.53	2.62±1.25	6.95±1.49	16.73±2.42

Table 1: Average ± Standard Deviation of RMSE of QFT reconstruction

From these results, we see that in order to maintain the same reconstruction error bound, quadrupling the input size (i.e. from 2x2 to 4x4) requires a little less than a 10x increase in the number of circuit simulations. In other words, the number of circuit runs is roughly linear with respect to image input size n. Combined with the $O(poly(log(n))$ time to set up and run the circuit, we see that simply decoding the qubit image state takes $O(npoly(log(n)))$ time, which not only is just as slow as the classical Fourier transform, but also now introduces some error into the reconstructed image. On the other hand, we do benefit from exponentially decreasing the necessary storage of the image, from n classical pixels to $log_2(n)$ qubits.

4.2 Sparse Signal Processing Attempt

In this section, we will attempt to subsample the frequency domain, to further improve image compression rates. One classical technique for reducing image storage size is to throw away the high frequency coefficients in the frequency domain, exploiting the relative sparsity of the frequency domain to achieve better image compression. Not only are the magnitudes of these high frequency coefficients generally very small for most "natural-looking" images, but differences between very high frequencies are often difficult to see visually.

For this approach, I wanted to see how removing the highest frequency Fourier and Hadamard coefficients would affect image reconstruction quality for both transforms. I simulated "zeroing out" qubits through Qiskit's reset gate, which simply performs a swap gate with an ancilla qubit initialized to $|0\rangle$. Because we store 2^n frequency coefficients within n qubits, by zeroing out some $k < n$ qubits, we are essentially only keeping 2^{n-k} coefficients, or in other words only $\frac{1}{2^k}$ of the original coefficients. On the other hand, this means we would only have to store k qubits per image, instead of n. Figure 7 shows an example circuit for a 4-qubit image, where we keep $\frac{2^2}{2^4} = 0.25$ of the Hadamard coefficients.

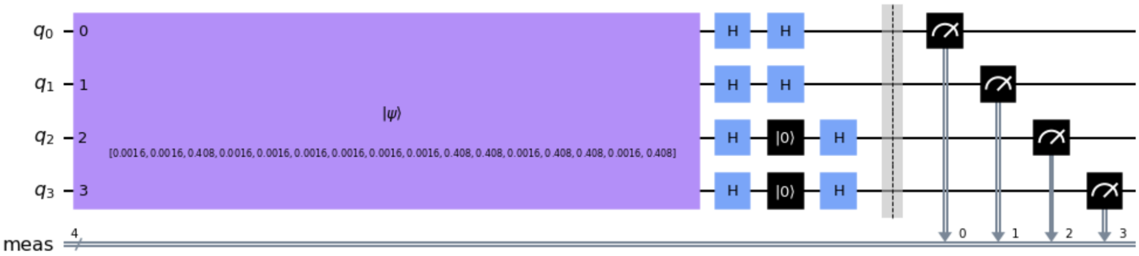


Figure 7: Zeroing 2 qubits of a 4 qubit image in the Hadamard basis. A similar technique can be done for the Fourier basis.

I decided to explore what percentage of coefficients we can remove without losing much too image quality. I decided to perform two separate sets of experiments: one on 100 random images of size 8x8 (which likely have significant high frequency coefficients due to the randomness of the image) and one on 100 downsampled 8x8 images from the MNIST database (because these images are relatively smooth they likely have more significant low frequency coefficients). For each individual image, I performed 100000 circuit simulations, so issues related to poor sampling of the output state will not affect these experiments. I varied the number of qubits I zeroed out and tried both the quantum Hadamard and quantum Fourier transforms. Numerical results are shown in Tables 2 and 3, and visual results can be seen in Figures 9 and 8.

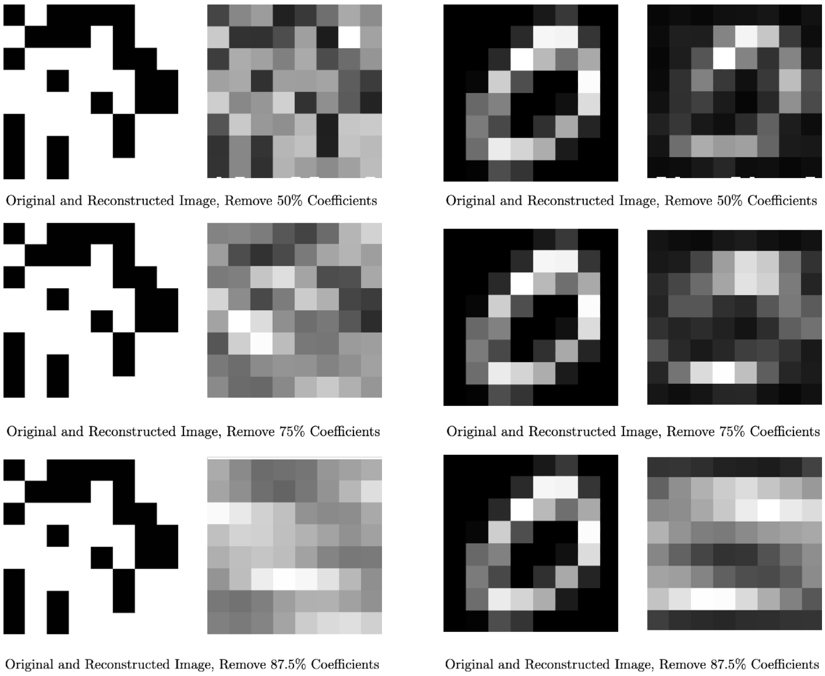
RMSE of 8x8 MNIST	Fourier	Hadamard
No image processing	27.65±3.47	27.65±3.47
Zero 1 qubit (remove 50% of frequency coefficients)	35.27±5.62	57.02±8.87
Zero 2 qubits (remove 75% of frequency coefficients)	56.79±10.03	78.97±13.83
Zero 3 qubits (remove 87.5% of frequency coefficients)	101.08±20.78	91.54±18.09

Table 2: Average ± Standard Deviation of RMSE of Reconstructed MNIST Images

RMSE of 8x8 Random Images	Fourier	Hadamard
No image processing	2.93±0.98	2.93±0.98
Zero 1 qubit (remove 50% of frequency coefficients)	78.71±5.53	90.11±7.90
Zero 2 qubits (remove 75% of frequency coefficients)	104.30±5.69	112.71±7.78
Zero 3 qubits (remove 87.5% of frequency coefficients)	121.11±8.22	127.72±9.96

Table 3: Average ± Standard Deviation of RMSE of Reconstructed Random Images

From these results, we see that removing the high frequency coefficients from the Fourier basis greatly impacts the quality of reconstruction, and removing high frequency coefficients from the



Original and Reconstructed Image, Remove 50% Coefficients

Original and Reconstructed Image, Remove 50% Coefficients

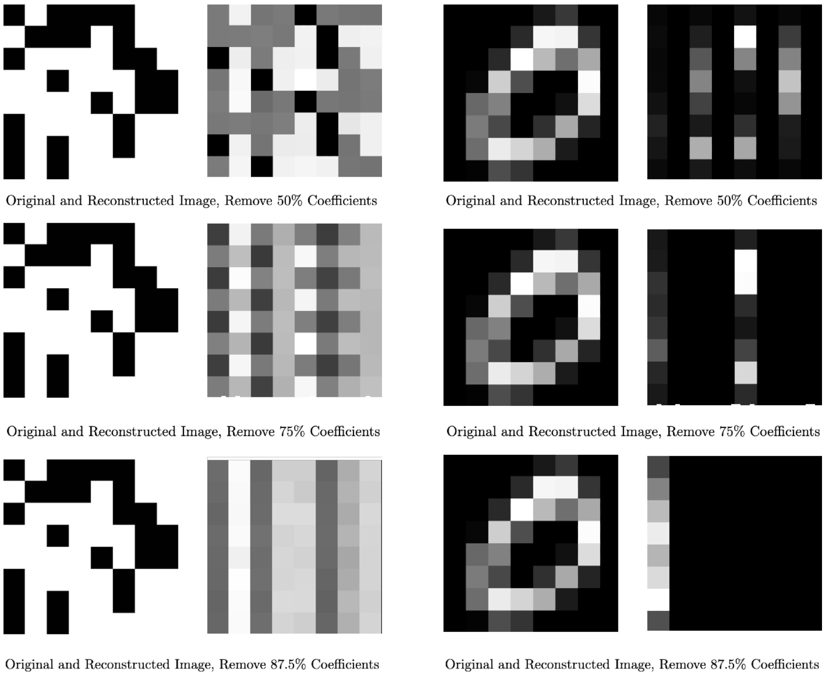
Original and Reconstructed Image, Remove 75% Coefficients

Original and Reconstructed Image, Remove 75% Coefficients

Original and Reconstructed Image, Remove 87.5% Coefficients

Original and Reconstructed Image, Remove 87.5% Coefficients

Figure 8: Reconstructed random and MNIST images w.r.t % Fourier coefficients removed



Original and Reconstructed Image, Remove 50% Coefficients

Original and Reconstructed Image, Remove 50% Coefficients

Original and Reconstructed Image, Remove 75% Coefficients

Original and Reconstructed Image, Remove 75% Coefficients

Original and Reconstructed Image, Remove 87.5% Coefficients

Original and Reconstructed Image, Remove 87.5% Coefficients

Figure 9: Reconstructed random and MNIST images w.r.t % Hadamard coefficients removed

Hadamard basis simply deletes parts of the image. As expected, within the Fourier reconstructions, the generally lower frequency MNIST images had significantly better reconstruction quality compared to the high frequency random images. But even removing just 50% of the coefficients leads to drastic losses in quality. Unfortunately, this corresponds to removing a single qubit of information, a constant space decrease, whereas the number of qubits necessary to store an image increases logarithmically with the the input image size. For the removal of 2 or more qubits' information, even the reconstructed low frequency images becomes unrecognizable. It is clear that this sparse signal processing approach is not well suited for quantum image processing.

5 Conclusion

Quantum image processing seems to be very promising in terms of space complexity, but accuracy and time complexity of classical image processing-inspired compression and reconstruction is still lacking. Even in the completely ideal world of Qiskit simulation, where there are no gate errors, simply decoding an 16x16 image from a qubit state requires thousands, if not hundreds of thousands of circuit simulations to accurately sample the reconstructed pixel values. Trying to sample the frequency domain leads to even worse reconstructions, except when the image is very low frequency, and even so we can only feasibly remove a constant number of qubits of information before ruining the reconstruction. Because the number of qubits scales logarithmically with input image size, to remove a constant number of qubits in this manner becomes useless for large inputs.

However, there is hope. We tried to compress the logarithmic qubit representation of an image, which already takes exponentially lower space than a classical bitwise representation of the same image. In addition, some quantum-inspired algorithms, such as computing image similarity and rotational symmetry, seem to vastly outperform any sort of classical approach, with an exponential speedup already being observed.

References

[1] Glenn Beach, Chris Lomont, and Charles Cohen. Quantum image processing (quip). In *32nd Applied Imagery Pattern Recognition Workshop, 2003. Proceedings.*, pages 39–44. IEEE, 2003.

[2] Xi-Wei Yao, Hengyan Wang, Zeyang Liao, Ming-Cheng Chen, Jian Pan, Jun Li, Kechao Zhang, Xingcheng Lin, Zhehui Wang, Zhihuang Luo, et al. Quantum image processing and its application to edge detection: theory and experiment. *Physical Review X*, 7(3):031041, 2017.

[3] Andrei N Soklakov and Rüdiger Schack. Efficient state preparation for a register of quantum bits. *Physical review A*, 73(1):012307, 2006.

[4] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[5] Qiskit. Quantum fourier transform, Apr 2022.