# QUANTUM HADAMARD EDGE DETECTION

MINOR PROJECT REPORT

By

**Sagnikta Nath Chaudhuri (Reg No. RA2211003011327) Madhu Shree Aravindan (Reg No. RA2211003011773) Devanshu Bohra (Reg No. RA2211003012048) Kamalesh Koiri (Reg No. RA2211003012082)**

Under the guidance of Dr. Raghavendra V

*In partial fulfilment for the Course*

of

**21CSE306J – QUANTUM COMPUTATION**

in **Department Of Computing Technologies**



**FACULTY OF ENGINEERING AND TECHNOLOGY SCHOOL OF COMPUTING**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR**

**MAY 2024**

****

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that this minor project report for the course 21CSE306J QUANTUM COMPUTATION entitled in **"QUANTUM HADAMARD EDGE DETECTION"** is the bonafide work of Sagnikta Nath Chaudhuri (Reg No. RA2211003011327), Madhu Shree Aravindan (Reg No. RA2211003011773), Devanshu Bohra (Reg No. RA2211003012048) and Kamalesh Koiri (Reg No. RA2211003012082) who carried out the work under my supervision.

**SIGNATURE**

Dr. Raghavendra V

**Assistant Professor**

**Computing Technologies Department** SRM Institute of Science and Technology Kattankulathur

**SIGNATURE**

Dr. M. PUSHPALATHA

**Professor and Head**

**Computing Technologies Department** SRM Institute of Science and Technology Kattankulathur

## ABSTRACT

In this project, we aim to address the image edge detection problem using quantum computing. Vision serves as a critical source of information, and image processing plays a pivotal role in enabling machines to interpret this visual data. As the significance of processing digital images continues to grow, classical image processing methods encounter challenges, particularly when dealing with large datasets.

To overcome these inefficiencies, we propose an algorithm that harnesses the power of quantum computing. By leveraging the unique properties of quantum systems, such as the utilization of Hadamard gates, we seek to develop a novel approach to image edge detection. This endeavor represents an innovative step towards enhancing the efficiency and scalability of image processing tasks.

## ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V. Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project.

We are highly thankful to our Course project Faculty **Dr. Raghavendra V, Assistant Professor, Department of Computing Technologies,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend our gratitude to our **HoD Dr.M. Pushpalatha, Professor Department of Computing Technologies** for her Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **CONTENTS** | **PAGE NO** |
| **1** | **INTRODUCTION**  1.1 Motivation | 8  8 |
|  | 1.2 Objective | 8 |
|  | 1.3 Problem Statement | 8 |
|  | 1.4 Challenges | 9 |
| **2** | **LITERATURE SURVEY** | 10 |
| **3** | **REQUIREMENT** | 11 |
| **4** | **METHODOLOGY** | 12 |
| **5** | **IMPLEMENTATION** | 16 |
| **6** | **RESULTS & DISCUSSION** | 22 |
| **7** | **CONCLUSION** | 24 |
| **8** | **REFERENCES** | 25 |

# 1. INTRODUCTION

An image can be defined as a two-dimensional function f(x, y), where x and y represent spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is known as the intensity or gray level of the image at that point. When x, y, and the amplitudes of f are all finite and discrete quantities, the image is termed a digital image.

Image processing involves the manipulation of digital images using various digital techniques. It is a crucial aspect of Computer Vision, a broader field concerned with processing and analyzing images. A digital image comprises a finite number of elements, each with a specific location and value known as pixels.

Among the numerous features of an image, edges are fundamental. Edge detection, a fundamental concept in image processing, aims to identify and locate the boundaries of objects within images. Accurate edge detection is essential for various image preprocessing tasks, including object detection, image segmentation, and image compression.

### Motivation

While classical methods such as the Sobel, Prewitt, Laplacian, and Canny edge detectors have long been employed for edge detection tasks, they can prove inefficient when applied to large images due to the computational load associated with processing numerous pixels. Quantum edge detection offers a promising alternative, utilizing its own unique methods to detect edges in images.

### Objective

To create a quantum edge detection algorithm using Hadamard gate to increase the efficiency of edge detection algorithm.

### Problem Statement

As previously mentioned, edge detection plays a critical role in image processing and is necessary for addressing many image processing tasks. However, traditional classical methods for edge detection can become inefficient as image sizes and datasets increase in scale. To mitigate this inefficiency, we propose a methodology that leverages the quantum Hadamard gate for edge detection.

### Challenges

The challenges encountered in image processing, spanning machine learning, linear filtering and convolution, multiscale analysis, face and pattern recognition, and image and video coding, can be effectively tackled through the utilization of the Quantum Hadamard edge detection algorithm. By encoding image and video information into qubits, this algorithm enables not only efficient processing but also secure transmission of data through networks safeguarded by quantum technology.

# LITERATURE SURVEY

Due to limitations faced in classical computing when processing large data for image processing, researchers and professors have explored methods for performing edge detection using quantum computing. One such work, by Yi Zhang, Kai Lu, and YingHui Gao [1], proposed an edge extraction algorithm based on the flexible representation of quantum images (FRQI) and the well-known Sobel edge extraction algorithm. This resulted in the design of a novel quantum edge extraction algorithm named QSobel. Initially, two unitary operations of the shift transformations of FRQI were designed. These operations were then utilized to calculate the intensity gradient of every pixel in the FRQI image simultaneously. Finally, the result of edge extraction was stored in a new FRQI image.

Several researchers have conducted comparative analyses of edge detection methods, as demonstrated in a study by Junfeng Jing, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun [2]. This study analyzed various traditional edge detection techniques such as the Roberts operator, Prewitt operator, Sobel operator, and Canny edge detection. Deep learning approaches, including CNNs, Backbone networks, AlexNet, and VGGs, were also discussed. When analyzing the results using specific metrics, deep learning methods like Canny and DeepEdge appeared to perform well, albeit with increased computational complexity. Many algorithms exhibited varying performance when applied to the same images under different lighting conditions.

Some researchers have focused on neural networks for edge detection due to their maximal accuracy. For instance, Artyom M. Grigoryan, Sos S. Agaian, and Karen Panetta [3] studied a new paired transform-based quantum representation and computation of one-dimensional and 2-D signals convolutions and gradients. They defined a new visual data representation to simplify convolution calculations, making it feasible to parallelize convolution and gradient operations for more efficient performance. The efficiency of this approach was demonstrated on multiple illustrative examples for quantum edge detection, gradients, and convolution.

Building upon these past studies, this project aims to create an efficient edge detection algorithm using the Hadamard gate. This choice is made as neural networks or deep learning methods may not be efficient. The Hadamard edge detection algorithm has shown efficient performance, and further tuning for a variety of images under different conditions could make it the best algorithm for edge detection so far.

1. **REQUIREMENTS**

**3.1 Software Requirements**

1. **Python Environment:** Python is required to execute the code.
2. **Qiskit Library:** Qiskit is a quantum computing library for Python, used extensively in the code.
3. **IBM Quantum Lab:** a cloud-based platform that provides access to quantum computing resources
4. **Matplotlib:** Matplotlib is used for plotting images in the code.
5. **Numpy:** Numpy is required for numerical computations and array manipulations in Python.
   1. **Hardware Requirements**
6. **Quantum Backend:** Access to a quantum backend, such as a simulator or real quantum device, is needed to execute quantum circuits.
7. **Classical Hardware:** A standard classical computer is required to run the Python code and perform classical post- processing tasks.
8. **Quantum Computers :** N/A

**3.3 Additional Considerations**

1. **Qiskit Runtime:** Access to Qiskit Runtime services may be required for executing quantum circuits.
2. **Backend Configuration:** Configuration of Qiskit with appropriate API tokens is necessary if using a real quantum backend

# METHODOLOGY

**4.1 Converting Classical Image to Quantum Image**

The first step is to input the image and normalize them and then converting them from classical to quantum images. This is done by Quantum Probability Image Encoding (QPIE) which uses the probability amplitudes of a quantum state to store the pixel values of a classical image. If we have n qubits, we have access to up to

2n states in superposition. In QPIE we take advantage of this fact to design an efficient and robust encoding scheme for Black-and-White (B&W) or RGB images and exponentially reduce the memory required to store the data. That means, for storing a 4-pixel image, we need just 2-qubits; for 8-pixel image we need 3-qubits, and so on. In general, the number of qubits (n) for an N-pixel image is calculated as:

n = [log2N]

A black and white squares with white text

Description automatically generated

Let’s take a sample image with four pixels which is arranged in 2D.

Here, the vector (I0, I1, I2, I3) represents color intensities (in 8-bit B&W color) of different pixels represented as a 2D matrix to form a

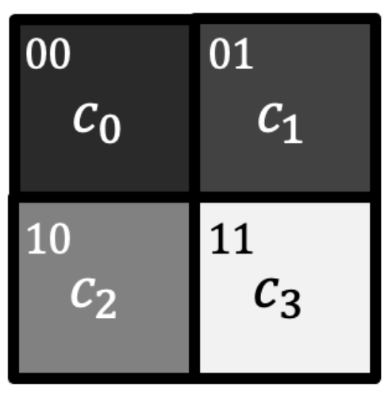
.intensities as follows:

I = (Iyx)N1XN2

Now, we need to represent these pixel intensities as the probability amplitudes of a particular quantum state. To do this, the pixel intensities should be normalized so that the sum of the squares of all the probability amplitudes is 1. For every ci corresponding to respective Iyx the normalization can be done as follows:

ci = Iyx/√∑I2yx

After Normalization,



Assigning the normalized pixel color values of each pixel Pi to the respective |i> quantum state, we can write the image state |Img> as:

|Img> = c0 |00> + c1 |01> + c2 |10> + c3 |11>

Such a state can be very efficiently prepared just by using a few rotation and CNOT gates.

A diagram of a circuit

Description automatically generated

**4.2 Quantum Hadamard Edge Detection (QHED)**

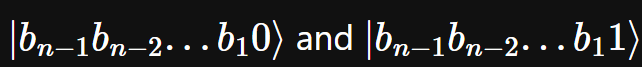
The Hadamard gate *H* has the following operation on the state of qubit,

|0> → ((|0> + |1>)/√2) |1> → (|0> - |1>)/√2

The QHED algorithm generalizes this action of *H-*gate and uses it for edge detection of an image.

Let us assume we have an N-pixel image. The pixels of the image can be numbered using binary bit-strings in the form of |bn-1 bn-2 bn-3…b1b0> where bi ↋ {0,1}.

For two neighboring pixels, only the least significant bit (LSB) is different for both,



Now, if we apply the *H*-gate to the LSB of an arbitrary size quantum register, we can represent the resultant unitary like,

A black background with white numbers and symbols

Description automatically generated

Applying this unitary to a quantum register containing pixel values encoded using the QPIE representation. We have,

A black background with white text

Description automatically generated

We add an extra auxiliary qubit to the register which we can utilize to extend the QHED algorithm and perform computation on both even and odd-pixel-pairs simultaneously.

Like the last time, we initialize to the state, however, the *H*-gate is now applied to the auxiliary qubit this time which is initialized to state |0>.

This produces an (n+1) qubit redundant image state which can be represented as,

A black background with white text

Description automatically generated

Since we get the redundant probability amplitudes obtained in the resultant state we can define an amplitude permutation unitary as follows to transform the amplitudes into a structure which will make it easier to calculate the image gradients further ahead,

A black and white screen with numbers and symbols

Description automatically generated

The above unitary corresponds to a **Decrement gate**. Hence, we can efficeintly decompose this unitary into a set of single- and multi-controlled-X rotations on a register of multiple qubits.

Now again if we apply the *H*-gate to the auxiliary qubit, we obtain the gradients for both even- and odd-pixel-pairs at the same time like so,

A black background with white text

Description automatically generated

Finally, measuring this state conditioned on the auxiliary qubit being in state |1>, we will get the resultant horizontal gradient values (ci – ci+1) for all possible pairs of adjacent qubits. This gives us the horizontal scan of the entire image.

The above process is repeated and the transpose of the matrix is taken, for vertical scan of the entire image. The horizontal and the vertical scan are then added to get the final edge detected.

1. **IMPLEMENTATION**

The code utilizes the Qiskit library in Python to demonstrate quantum image processing. It begins by representing an 8x8 binary image as a numpy array and then converts the raw pixel values to probability amplitudes using amplitude encoding. Quantum circuits are constructed to perform horizontal and vertical scans on the image. The circuits are simulated using the statevector simulator backend from Qiskit Aer. Classical post-processing is applied to generate edge-detected images by thresholding the output statevectors. Finally, the original and edge-detected images are plotted using Matplotlib.

A white rectangular object with a black border

Description automatically generatedA screenshot of a computer program

Description automatically generated

For the purpose of this demonstration, we can assume that an image is nothing but a collection of pixel values represented as a numpy matrix in python.

A screenshot of a computer

Description automatically generated

Output:

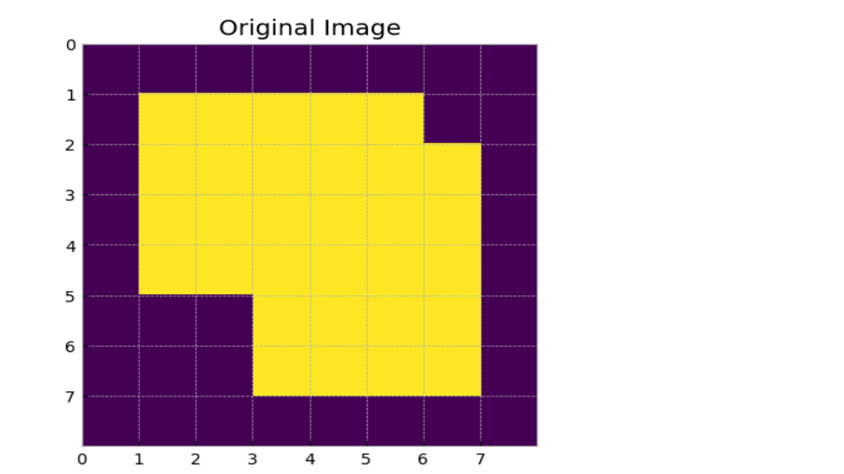


Image is created using Numpy array for demonstration purpose and to get cleaner image.

A screenshot of a computer program

Description automatically generated

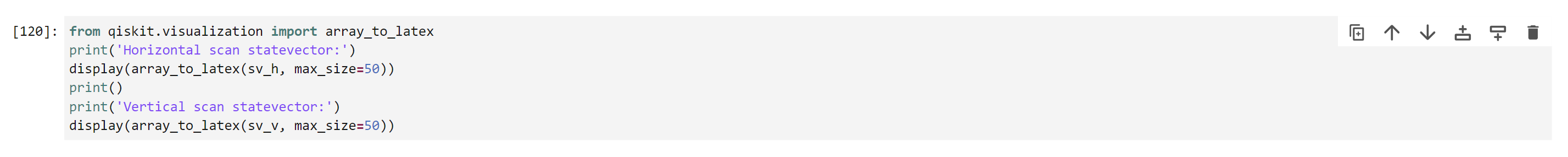
As can be seen in the above python code, we obtain two different amplitude encoded quantum images. The first one (image\_norm\_h) is for the horizontal scanning of the image and the second one (image\_norm\_v) is for the vertical scanning of the image.

After this we initialize the number of qubits and the amplitude permutation unitary

A screenshot of a computer program

Description automatically generated

After normalizing and converting pixel values to probability amplitudes, the quantum circuit is constructed. Using the initialize() method, the image is prepared as quantum states. Hadamard gates are applied to the auxiliary qubit, followed by the amplitude permutation unitary, and another Hadamard gate. This process is repeated for vertical scanning.



A close-up of a number

Description automatically generated

A screenshot of a computer program

Description automatically generated

We can clearly see that we need to consider only those states where the auxiliary qubit (qubit-0 or LSB in our case) gives a measurement output of ||1⟩. Since, we know that LSB is 1 in a bit-string only for odd numbers, we easily just take the amplitudes corresponding to odd states from the statevector to form our image and discard all the even states.

The following code, performs this task along with some classical post-processing to ensure that we get the best results when we plot our image. After we filter the required states from the raw statevector, we can rearrange the 1D array of amplitudes to a 2D matrix to get our edge detected horizontal and vertical scans like so,

A computer screen shot of a computer code

Description automatically generatedA white background with black text

Description automatically generated

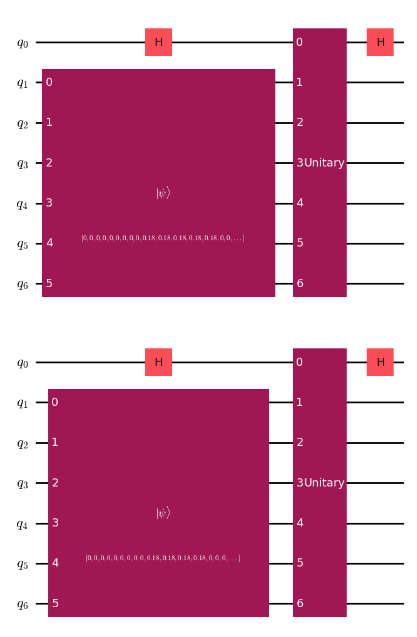
**Cell 1, 2,3:**

A yellow and purple square

Description automatically generated

With the image defined for testing, we encode the pixel intensities as probability amplitudes of different states in the system.

**Cell 4,5,6:**



The output consists of quantum circuits visualized using Matplotlib, illustrating the preparation of the image as quantum states and the application of amplitude permutation unitary operations for horizontal and vertical scanning.

**Cell 7,8,9:**

A screenshot of a graph

Description automatically generated

The output consists of edge-detected images obtained from horizontal and vertical scans of the original image,

where edges are highlighted by thresholding the raw statevector data.

**Cell 10:A yellow and purple square

Description automatically generated**

**A yellow and purple square with a letter

Description automatically generated**

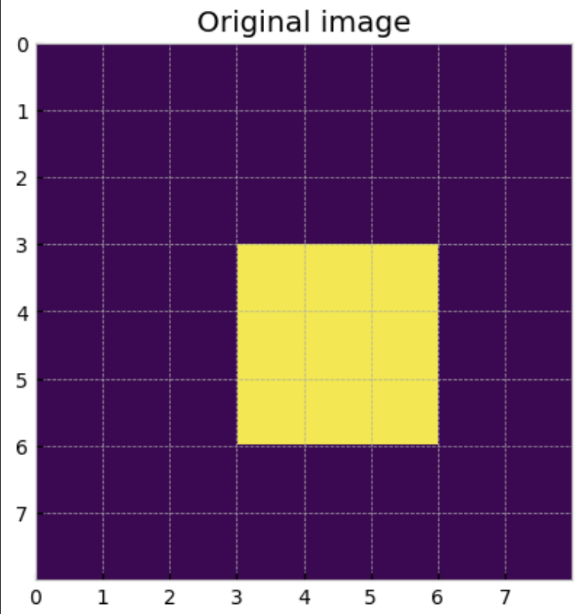
The output includes the original image and the combined edge-detected image obtained by merging the horizontal and vertical edge scan results.

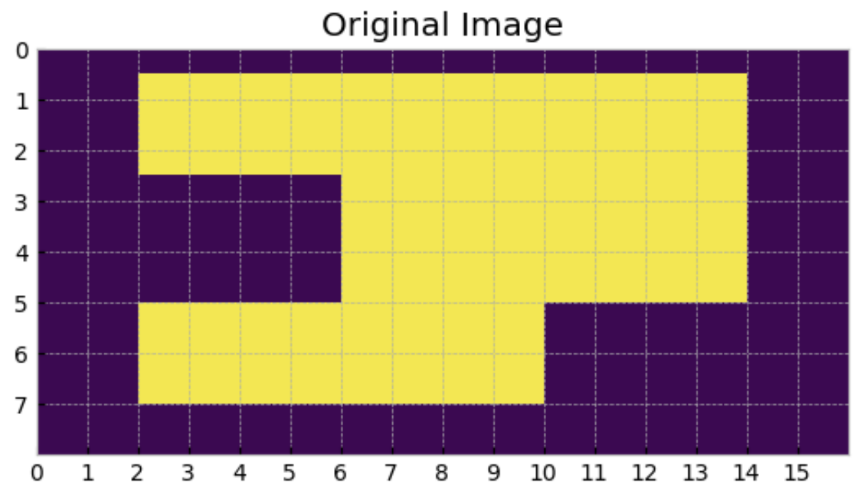
1. **RESULT AND DISCUSSION**

Quantum Hadamard Edge detection algorithm were applied for different shapes and were analyzed,

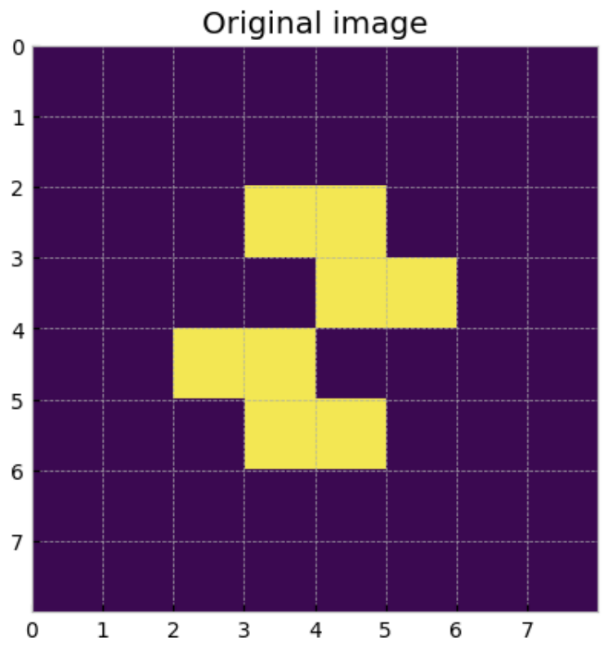
A purple and yellow square with a square in the middle

Description automatically generated

A yellow square with white text

Description automatically generated

A yellow and purple grid with white text

Description automatically generated

We can observe from the above results that the QHED algorithm is not very accurate, especially when the images get more and more complex the edge detected is also getting deferred. This can be improved by making modifications in the algorithm. Apart from that the time complexity has been significantly reduced, classical algorithms take O(2n) time to run where as QHED can achieve a time complexity of O(1).

A purple rectangular object with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**No. of Single Qubit Gate:** 2 (for each)

**No. of Two Qubit Gate:** 0

**Circuit Complexity:** 2 (two Hadamard gates for each)

**Query Complexity:** total number of qubits required (total\_qb)

1. **CONCLUSION**

In summary, Quantum Edge Detection emerges as a transformative approach at the intersection of quantum mechanics and image analysis. It offers unprecedented precision and speed by leveraging quantum principles, promising breakthroughs in fields like medical imaging, autonomous systems, and AI. While challenges exist, such as mastering quantum complexity and algorithm design, the potential rewards are immense. This innovative fusion opens new avenues for image understanding, where pixels and quantum particles unite to reshape industries and push the boundaries of visual comprehension. As quantum technologies advance, Quantum Edge Detection’s impact is poised to expand, unveiling a future where images reveal deeper insights than ever before.

# REFERENCES

**[1]** Y. Zhang, K. Lu, and Y. Gao, Qsobel: A Novel Quantum Image Edge Extraction Algorithm, Sci. China Inf. Sci. 58, 1-13 (2015).

**[2]** Junfeng Jing, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun. Recent advances on image edge detection: A comprehensive review. Neurocomputing, 503:259–271, 2022.

**[3]** Quantum-inspired edge detection algorithms implementation using new dynamic visual data representation and short-length convolution computation. arXiv.org, 2022.

**[4]** <https://medium.com/mit-6-s089-intro-to-quantum-computing/quantum-image-processing-quantum-probability-image-encoding-qpie-and-quantum-hadamard-edge-df7bd3dc7f8>

**[5]** <https://arxiv.org/pdf/1801.01465.pdf>

**[6]** <https://qiskit.org/textbook/ch-applications/quantum-edge-detection.html>