Distorted Edge Feature Extraction using Quantum Convolutional Structure

Yayu Mo
Electrical and Computer Engineering
Southern Methodist University (SMU)
Dallas, Texas, USA
yayum@smu.edu

Sanjaya Lohani
Electrical and Computer Engineering
Southern Methodist University (SMU)
Dallas, Texas, USA
slohani@smu.edu

Mitchell A. Thornton
Darwin Deason Institute for Cyber Security
Southern Methodist University (SMU)
Dallas, Texas, USA
mitch@smu.edu

***Abstract*—Advancements in quantum information have significantly impacted the field of image processing, although challenges remain. In edge detection, image distortion often hinders the extraction of target boundaries. In this paper, we propose a method to extract distorted edge features by applying shallow layers in quantum convolutional neural networks (QCNN). By combining the advantages of quantum computing and the layered structure of convolutional neural networks (CNN), this approach addresses the problem and compares the extracted distorted pattern with the reference pattern, achieving a best matching ratio of 99.71% in images interfered with impulse noise. Furthermore, we also compare the performance with the classical method and other quantum algorithms, our method gains a 97.19% ratio.**

***Index Terms*—edge detection, quantum convolutional structure, Cascade Controlled-X structure**

I. Introduction

Image processing techniques, such as edge detection, denoising, and deblurring, are of great significance for further classification and target detection tasks. Image processing and classical post-processing methods exhibit notable inefficiencies in the context of the big data era [1–3], which requires the development of new algorithms to enhance their performance. Quantum computing, with its advantages stemming from phenomena such as superposition and entanglement, has garnered significant attention from researchers. For example, the application of superposition in molecular biology addresses the dimensionality problem inherent in classical computers [4]. Furthermore, the benefits enable quantum communication, allowing for untraceable teleportation[5] and free-space links [6]. Furthermore, quantum computing is expected to revolutionize the security protocols in cryptography [7] as well as in the finance sector [8]. The potential of parallelization [9] in quantum computing facilitates the further study of image processing using quantum algorithms to address efficiency challenges.

Numerous studies have been conducted to verify the effectiveness of quantum algorithms in image processing. In edge detection, Yao et al. [10] applied the Hadamard Transformation to detect edges and compared the known pattern with the input pattern using the SWAP test. However, the experiments involving the SWAP test primarily considered known patterns such as rotation, without testing on images containing deformation or distortion, which are common in images captured by physical sensors or historical photographs.

Furthermore, some of the classical edge detection algorithms have been updated in quantum form. Sundani et al. [11] proposed a quantum version of the Canny algorithm, which detects not only sharp and clear edge features, but also weak ones, achieving a performance improvement of 4.05% over the classical Canny algorithm. Additionally, Chetia et al.[12] also proposed a quantum-enhanced Sobel edge detection algorithm to mitigate the effects of discontinuity and roughness.

However, few algorithms have considered the quantum form of CNN. In 2020, Li et al. [13] proposed a quantum deep convolutional neural network to address classification problems and compared its performance with traditional CNN. The verification of the QCNN's advantage over traditional CNN offers a new perspective for solving the challenges of extracting distorted edge features in the field of edge detection.

In this paper, we aim to extract the deformed edge feature by leveraging the advantages of shallow layers in QCNN to address deformation challenges. Unlike the entire structure, which includes the pooling layer, fully connected layer, and classifiers, the extraction of edge features requires only the feature map generated by the initial convolutional layers in the QCNN model. Our objective is to compare the QCNN approach with the normal pattern using the Cascade Controlled-X structure to demonstrate its effectiveness in solving deformation issues in image processing. The main workflow of our model, which takes an example of 2×2 convolutional circuit, is shown as Fig. 1. Additionally, comparisons with classical methods and other existing algorithms have been presented, illustrating our proof-of-concept and the improvements achieved.

The article is organized in the following sections. In Section II, we briefly introduce the backgrounds for the reader to understand some basic concepts in Quantum Edge Detection and Quantum Convolutional Neural Networks. In Section III, we introduce our approach to solve the problem and the dataset in our experiments. In Section IV, we present our results. After that in Section V, we summarize the results and give a discussion for our future works.

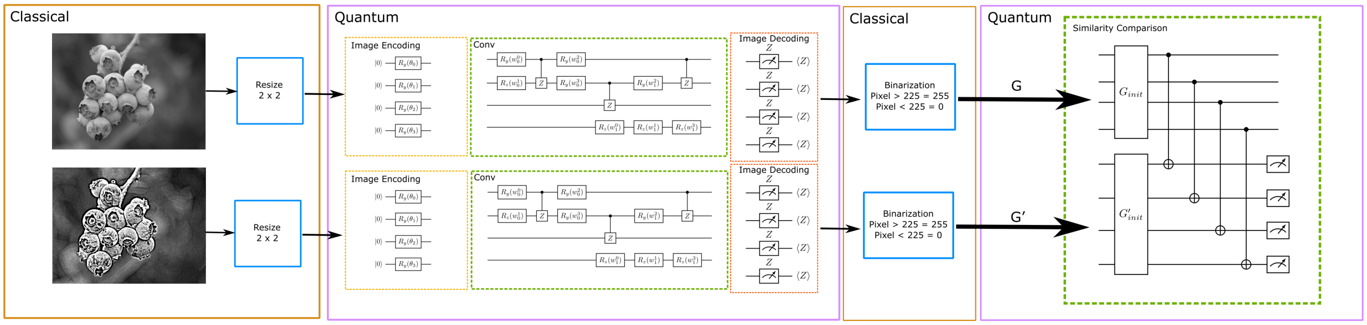


Fig. 1: The model consists of two parts: quantum and classical. The quantum part includes the convolutional circuit and similarity comparison circuit. The classical part includes image resizing and binarization.

II. Background

In this section, we provide a brief overview of basic quantum notations, gates, and a quantum convolutional layer to facilitate understanding of the concepts of image processing using QCNN. Additionally, we introduce the comparison circuit based on the Control NOT (CNOT) gate.

A. Quantum Notations

The mathematical form of quantum computation is written in Dirac Notations $|.\rangle$ (and dual form $\langle.|$) [14]. In most circumstances, to represent the quantum state of a single qubit, the computational basis is introduced. The most general basis is $\{|0\rangle, |1\rangle\}$, which are explicity written as,

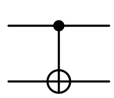
$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}. \tag{1}$$

An arbitrary single qubit quantum state is expressed as the combination of computational basis with the probability amplitude α and β, which is denoted as

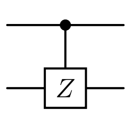
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \equiv \begin{bmatrix}\alpha\\\beta\end{bmatrix} \tag{2}$$

B. Quantum Gates

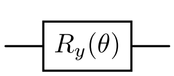
Quantum gates are the fundamental components of quantum computers, performing logical operations on qubits. The gates that we implement in our model are expressed as follows,



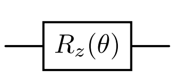
$$= \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix},$$



$$= \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$$



$$= \begin{bmatrix}cos\frac{\theta}{2} & -sin\frac{\theta}{2}\\ sin\frac{\theta}{2} & cos\frac{\theta}{2}\end{bmatrix}$$



$$= \begin{bmatrix}e^{i\frac{\theta}{2}} & 0\\ 0 & e^{-i\frac{\theta}{2}}\end{bmatrix}.$$

The gates from the top to bottom, respectively, represent the CNOT gate, control-Z, rotation about y-axis (R_y), and rotation about z-axis (R_z). Note that, we use the control-Z, R_y and R_z gates in the image encoding and QCNN architecture, which are described in the following sections.

C. Quantum Convolutional Layer

We implement a quantum convolutional layer structure using the Random Layer subroutine, a type of general quantum variational circuit [15]. For example, the diagrams of a 2×2 convolution kernel circuit are shown in Fig. 2a, comprising 8 rotation gates and 3 control-Z gates. In the decoding phase, all four qubits are measured using the Pauli-Z observable basis, enabling the calculation of the expectation value representing four copies of the convolved value after the 2×2 convolutional operation, as illustrated in Fig. 2b.

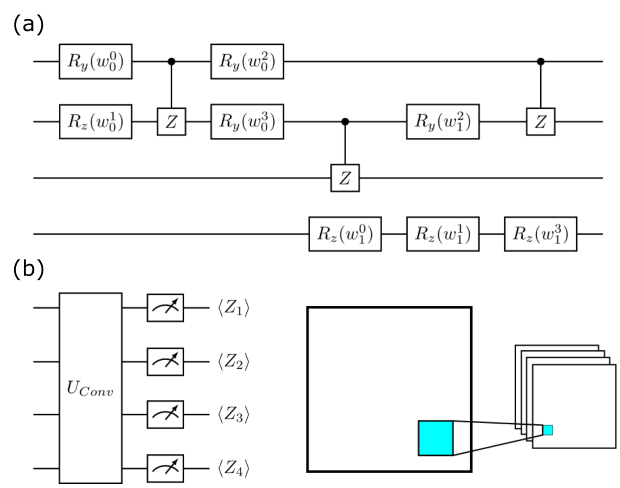


Fig. 2: (a) An example of a 2×2 convolution kernel circuit. (b) A diagram of decoding circuit of 4 qubits, where U_{conv} indicates the convolutional unit. The four outputs represent the copies of the convolved value from 4 inputs.

D. Qubit Comparison based on CNOT Gate

The CNOT gate can also be applied to compare base state qubits by measuring the second qubit, as illustrated in Fig.3.

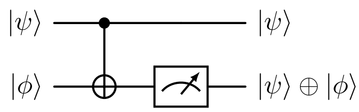


Fig. 3: Control NOT circuit

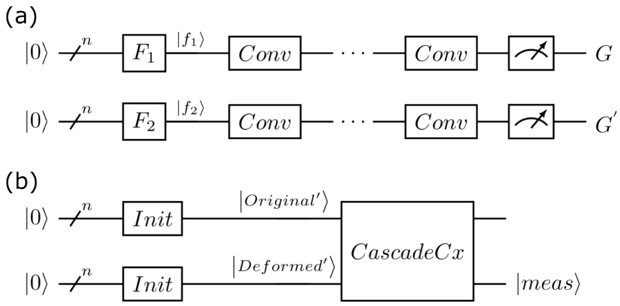


Fig. 4: (a) Brief diagram of Image Processing Circuit, where loops are needed to convolve the entire image. (b) Image comparison circuit diagram.

The evolution of the second qubit can be recognized as an XOR gate in a classical circuit. This implies if $|\psi\rangle = |\phi\rangle$, the final state being measured must be $|0\rangle$; otherwise, the measure will yield $|1\rangle$.

III. Approach

A. Model

In classical convolutional neural networks, feature maps generated by shallow layers usually contain detailed information, such as edge feature. Inspired by classical methods, this paper proposes two models as shown in Fig. 4, including the following functions: (1) Image encoding, feature map generation and decoding; (2) Classical binarization; and (3) Binary image comparison. Also, the image encoding part is shown as F_1 and F_2, representing the reference image and distorted image, respectively. The specific details are introduced as follows:

Image Encoding. To implement image processing in a quantum machine, encoding images into specific forms that can be stored in a series of quantum registers is the most important part. In the model, we apply general angle encoding, which is also known as qubit lattice image representation [16], to process in QCNN. Given an image $[F_{i,j}]_{W\times H}$, the quantum state $|f\rangle$ can be mapped as

$$|f\rangle = \otimes_{i,j}^{W\times H} \cos(\theta_{i,j})\,|0\rangle + \sin(\theta_{i,j})\,|1\rangle\,. \tag{3}$$

Where it can theoretically encode an image with the dimensions of $width(W) \times height(H)$, and θ represents the

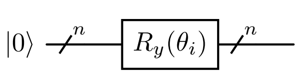
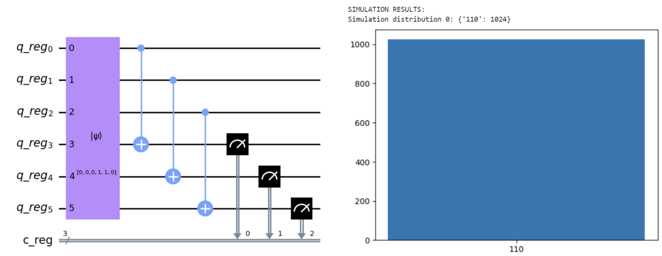


Fig. 5: Angle encoding circuit



(a) cascade Cx (b) simulation

Fig. 6: Cascade control-NOT structure with 3×2 qubit and corresponding simulation result.

normalized pixel value within the range $[0, \frac{\pi}{2}]$. For clarity, the value of $\theta_{i,j}$ can be formulated as follows,

$$\theta_{i,j} = \frac{F_{i,j} * \pi}{255 * 2}. \tag{4}$$

In our approach, we implement encoding through the Rotation-Y gate, as discussed in the background section and illustrated in Fig. 5. The parameter θ_i represents each pixel being encoded. We encode distorted images and reference images in two groups to ensure they are applied in the same model, as the model may vary when different initial weights are used in the convolutional layers.

Feature Map Generating and Decoding. In the feature map generation part, inspired by the convolutional layer in the renowned classical CNN, MobileNetV2 [17], we implement the quantum 3×3 convolutional circuit with a stride of 2. We also modify the input size from 224×224, which is the original input size in MobileNetV2, to 33×33, necessitating extra padding of 1 since we set our image size to 32×32 to fit the performance on a single PC. To generate the feature map, we apply N layers to extract the feature, where $N = 1, 2, 3$ (more than 4 layers could include more semantic information that is redundant for edge detection), and adjust it through the measurement of image comparison. In Fig. 4, the $Conv$ part represents a block of loops to perform the convolution operation throughout the image since a convolutional circuit can output only one value. We also set the seed to default to avoid randomly generating circuits in each loop, ensuring that our experimental results are consistent.

Binarization. Prior to engaging in the pattern comparison phase, image binarization must be applied to meet the requirements of the binary input in the comparison circuit. For convenience, we inverted the pixel value of the image and then applied the embedded threshold function supported by OpenCV. This approach has the added advantage of making the object shape much clearer with a high threshold.

Comparing Pattern. In pattern comparison, a Cascade CNOT structure is applied to compare the extracted specific features. As shown in Fig. 4b and Fig. 6a, the initialization part primarily consists of Pauli-X gates when the input pixel value is 255. After measurements, we only consider the output state $|meas\rangle$, as shown in Fig. 6b.

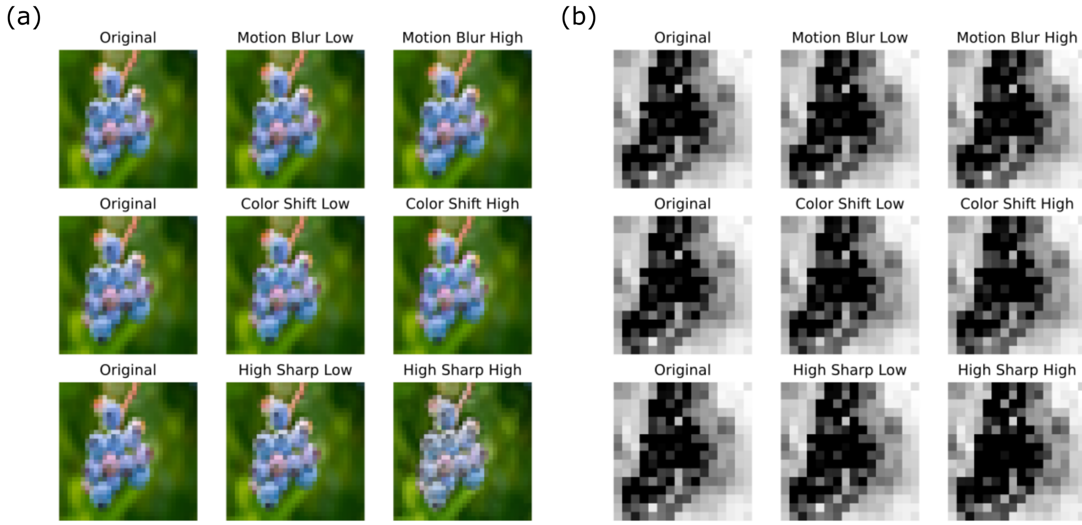


Fig. 7: (a) Resized images due to the requirements of image encoding. (b) Feature map visualization extracted by quantum convolutional layer.

After measurement, we calculate the similarity ratio (SR) using the equation,

$$SR = \frac{number\ of\ 0s\ in\ |meas\rangle}{length(|meas\rangle)}, \tag{5}$$

where the similarity ratio represents the similarity between the edge pattern of the distorted image and the reference image. The SR value illustrates the capability of the model to detect distorted edges.

B. Datasets

The KADID-10k dataset [18] comprises 81 reference images and includes 7 categories of distortions, including blurs, color distortions, compression, noise, brightness change, spatial distortions, and sharpness and contrast. Each image is subjected to 25 distortion types with 5 levels to represent the intensity of distortions. With a total of 10,125 distorted images, this dataset can be utilized to verify our model's robust capability to extract various types of distortion feature.

IV. Result

In the image encoding, we resize each image from the size of (512, 384) to (32, 32), as shown in Fig 7a. After encoding the image to amplitude information in the Rotation-Y gate, we exploit a single 3×3 quantum convolutional circuit with a single padding and stride length of 2 to generate the feature map and then decode them at the output. The total of 9 outputs are obtained, however, we only choose the best one for the visualization, shown as Fig. 7b.

We can see from the figure that it is evident that the basic shape features are included. Then we apply the distorted image and reference image to the binarization operation in OpenCV to generate the binary images containing only pixel values of 0 and 255. Since the binary images are best fit for the functions of Cascade CNOT gates mentioned earlier, we re-encode the image to a quantum state, where the state vector contains only the probability amplitude of 0 and 1. The binary images of the reference image and high sharp distortion image, along with a simply subtracted image, are shown in Fig. 8.

Fig. 8: Comparison between reference image and high sharp distortion image in level 5.

After going through the comparison circuit, we measure the results and calculate the pixel similarity ratio, as shown in Fig. 9 (green color). As the level increases, the capability would drop, but it still maintains a high similarity ratio close to 93% for the level 5 distortion. In addition, we compare existing edge detection methods, including classical Sobel [19], Canny [20], the Quantum Hadamard Edge Detection Method (QHED) [10], and the classical CNN-based Unet image segmentation model [21] with our QCNN-based model on five levels of high-sharpen image. To ensure that they share the same experimental settings, we apply the same image dimension of 16×16. Note that we implement a pre-trained Unet model from the Tensorflow library [22]. The Unet output is further downgraded to the size of 16×16 to keep the same experimental settings. All the outputs of different methods

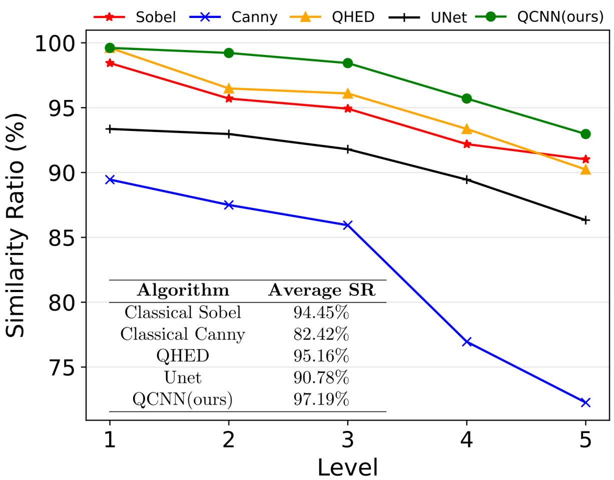


Fig. 9: Comparison with existing methods using high-sharpen distorted image with five levels.

go through the binarization and comparison procedure in our quantum circuit; the results are shown in Fig. 9. Note that we obtain an average similarity ratio of 97.19%, as shown in the inset table, indicating that our method is highly competitive among the existing methods.

Furthermore, we conduct several evaluations on various types of distortions and noise to assess the robustness and effectiveness of our method. These evaluations include tests on blurs, color distortions, compression artifacts, noise, brightness changes, spatial distortions, and variations in sharpness and contrast. The comprehensive results of these evaluations are presented in TABLE I, demonstrating the method's capability to handle a wide range of distortion types and maintain high performance across different scenarios.

From the table, it is evident that our method is robust in detecting the edges of blurred, compressed, noised, spatially distorted, high-sharpness, and high-contrast images. However, it still exhibits limited performance on color-distorted and brightness-changed images. This indicates that while our approach is effective for a wide range of distortions, there are specific areas, such as color and brightness variations, where further improvements are needed to enhance the overall robustness and accuracy of the method.

V. Discussion

In this paper, we present an approach inspired by the classical convolutional layer to combine the quantum convolutional layer with edge feature extraction. This method leverages quantum computing and feature maps generated by shallow layers to extract edge features. Additionally, we apply an image pixel comparison structure for binary images to compare the distorted pattern with the reference pattern, verifying the model's ability to extract edge features. Our test results analyze the relationship between the similarity ratio and distortion intensity levels, and we evaluate the compatibility of applying different types of distortion to the image. We

TABLE I: Distortion types

Category	Type	AvgSR
Blur	Gaussian blur	98.43%
	Lens blur	97.98%
	Motion blur	99.24%
Color distortions	Color diffusion	88.36%
	Color shift	98.61%
	Color quantization	95.82%
	Color saturation 1	84.10%
	Color saturation 2	91.26%
Compression	JPEG2000	99.21%
	JPEG	99.25%
Noise	White noise	99.60%
	White noise in color component	99.60%
	Impulse noise	99.71%
	Multiplicative noise	99.61%
	Denoise	97.49%
Brightness change	Brighten	73.33%
	Darken	83.76%
	Mean shift	92.28%
Spatial distortions	Jitter	97.15%
	Non-eccentricity patch	99.29%
	Pixelate	98.66%
	Quantization	93.75%
	Color block	98.39%
Sharpness and contrast	High sharpen	94.48%
	Contrast change	97.29%

achieve a high average similarity ratio ranging from 73.33% to 99.71%. Furthermore, our method demonstrates an average similarity ratio of 97.19% compared to other existing edge detection algorithms. However, due to the computational load on a single PC, several aspects remain unexplored.

First, experiments on a larger scale are necessary. Edge features are clearer on a larger scale, as seen in many classical CNN models with large input sizes, such as the original input size of 214×214 in MobileNetV2. Additionally, convolutional kernel size, strides, and paddings could significantly impact the modeling of the QCNN circuit. Secondly, experiments on 3-channel images are required. Our current experiments are based on grayscale images, which are essentially single-channel images. However, different color spaces such as RGB, YCbCr, and HSV exist, and the quantum encoded values vary in these color spaces, potentially generating different results.

Our QCNN approach demonstrates superior performance in low-scale feature extraction compared to some classical methods in simulation. However, as the image dimension increases, implementing our method in physical devices becomes challenging due to the significant increase in the depth of the quantum circuit and the number of qubits required. Further exploration is needed in this area. Additionally, in this research work, we compare the performance of QCNN with the classical Unet-based edge detection approach. Comparing the performance of QCNN with other classical CNN or deep learning-based methods remains a broad field for future research. In our future work, we will focus on addressing the aforementioned challenges and limitations to further verify the performance of our model.

REFERENCES

[1] Abhay Kumar Bhadani and Dhanya Jothimani. Big data: challenges, opportunities, and realities. *Effective big data management and opportunities for implementation*, pages 1–24, 2016.

[2] S Vahini Ezhilraman and Sujatha Srinivasan. State of the art in image processing & big data analytics: issues and challenges. *Int J Eng Technol*, 7(2.33):195–9, 2018.

[3] Sanjaya Lohani, Joseph M Lukens, Atiyya A Davis, Amirali Khannejad, Sangita Regmi, Daniel E Jones, Ryan T Glasser, Thomas A Searles, and Brian T Kirby. Demonstration of machine-learning-enhanced bayesian quantum state estimation. *New Journal of Physics*, 25(8):083009, 2023.

[4] Alberto Baiardi, Matthias Christandl, and Markus Reiher. Quantum computing for molecular biology. *ChemBioChem*, 24(13):e202300120, 2023.

[5] Nicolas Gisin and Rob Thew. Quantum communication. *Nature photonics*, 1(3):165–171, 2007.

[6] Sanjaya Lohani and Ryan T Glasser. Coherent optical communications enhanced by machine intelligence. *Machine Learning: Science and Technology*, 1(3):035006, 2020.

[7] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Reviews of modern physics*, 74(1):145, 2002.

[8] Daniel J Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering*, 1:1–24, 2020.

[9] Surya Teja Marella and Hemanth Sai Kumar Parisa. Introduction to quantum computing. *Quantum Computing and Communications*, 2020.

[10] 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.

[11] Dini Sundani, Sigit Widiyanto, Yuli Karyanti, and Dini Tri Wardani. Identification of image edge using quantum canny edge detection algorithm. *Journal of ICT research and applications*, 13(2):133–144, 2019.

[12] Rajib Chetia, SMB Boruah, and PP Sahu. Quantum image edge detection using improved sobel mask based on neqr. *Quantum Information Processing*, 20:1–25, 2021.

[13] YaoChong Li, Ri-Gui Zhou, RuQing Xu, Jia Luo, and WenWen Hu. A quantum deep convolutional neural network for image recognition. *Quantum Science and Technology*, 5(4):044003, 2020.

[14] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[15] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1):2, 2020.

[16] Salvador E Venegas-Andraca and Sougato Bose. Storing, processing, and retrieving an image using quantum mechanics. In *Quantum information and computation*, volume 5105, pages 137–147. SPIE, 2003.

[17] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[18] Hanhe Lin, Vlad Hosu, and Dietmar Saupe. Kadid-10k: A large-scale artificially distorted iqa database. In *2019 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2019.

[19] Irwin Sobel. History and definition of the sobel operator. *Retrieved from the World Wide Web*, 1505, 2014.

[20] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[22] Tensorflow documentation — image segmentation. https://www.tensorflow.org/tutorials/images/segmentation. Accessed: 2024-08-16.