Applied Soft Computing Journal 185 (2025) 113866

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Quantum fuzzy logic for edge detection: A demonstration on NISQ hardware

G. Nunziata [a], S. Crisci [a], G. De Gregorio [a,b], R. Schiattarella [c,b,*], G. Acampora [c,b], L. Coraggio [a,b], N. Itaco [a,b]

[a] Dipartimento di Matematica e Fisica, Università degli Studi della Campania "Luigi Vanvitelli", Viale Abramo Lincoln 5, Caserta, I-81100, Italy
[b] Istituto Nazionale di Fisica Nucleare, Complesso Universitario di Monte S. Angelo, Via Cintia, Napoli, I-80126, Italy
[c] Dipartimento di Fisica, Università degli Studi di Napoli Federico II, Via Cintia, Napoli, I-80126, Italy

ARTICLE INFO

Dataset link: https://github.com/gnunziata42/QFED.git

Keywords:
Fuzzy logic
Edge detection
Quantum computing
Quantum fuzzy inference engine

ABSTRACT

Quantum computing offers the potential to enhance computational efficiency beyond classical methods, but practical implementation remains challenging due to the limitations of Noisy Intermediate-Scale Quantum (NISQ) hardware, namely, restricted qubit counts, limited connectivity, and the presence of noise and decoherence. This study presents a novel approach to edge detection by leveraging a recently developed Quantum Fuzzy Inference Engine, implemented on a NISQ device. We introduce an optimized quantum circuit for its implementation, reducing qubit requirements and gate depth to improve execution on NISQ hardware. To overcome constraints related to large-scale image processing, a hybrid quantum–classical lookup table approach is employed. Edge detection performance is evaluated on the Berkeley Segmentation Data Set and Benchmarks 500 dataset under different conditions, including classical execution, ideal quantum simulation, noisy quantum simulation, and NISQ hardware calculation. Results demonstrate that the quantum fuzzy logic-based edge detection achieves outcomes comparable to classical methods by using fewer operations, marking a step toward practical quantum-enhanced image processing.

1. Introduction

Quantum computing is rapidly emerging as a transformative technology with the potential to revolutionize various fields, including image processing and computer vision [1]. This paper explores the intersection of this innovative computational paradigm and image analysis, proposing a novel quantum edge detection algorithm.

Edge detection (ED) is a fundamental task in image processing that underpins various computer vision applications, including object recognition [2], image segmentation [3], and feature extraction [4]. While classical ED algorithms yield reliable results, their computational efficiency is inherently constrained by the limitations of classical hardware. In fields like medical imaging or autonomous navigation where image data continues to grow in both size and complexity, the demand for more efficient and scalable ED techniques becomes increasingly pressing. In principle, quantum computing offers a compelling alternative by enabling computational speedups that classical image processing methods cannot achieve [1,5]. However, despite its promise, the practical implementation of quantum algorithms for image processing and ED with real-world images remains extremely challenging due to the limitations of contemporary NISQ hardware [6]. These devices encounter issues such as brief coherence times, gate errors, and constrained qubit connectivity, all of which adversely affect their reliability and scalability. Consequently, there is a substantial disparity between the theoretical potential of quantum computation in image analysis and the computational capability exhibited by contemporary NISQ devices. To address this discrepancy, this paper introduces the very first quantum fuzzy-based ED mechanism that can be executed on real NISQ computers. It is noteworthy that the integration of fuzzy logic within classical ED algorithms is a well-established methodology. The capacity of fuzzy logic to interpret data through degrees of truth, as opposed to binary classifications, renders this soft computing approach particularly well-suited for managing the ambiguities inherent in image analysis [7]. A prevalent technique for executing ED through the implementation of fuzzy systems involves the use of decision-making agents that employ linguistic If-Then rules to identify edges [8–10].

Notably, a quantum analog of classical fuzzy inference systems was recently proposed [11]. This Quantum Fuzzy Inference Engine (QFIE) exhibits exponential acceleration in executing fuzzy rules compared to its classical counterparts. Nevertheless, the quantum circuit necessary to implement the algorithm is generally impractical to run on current NISQ devices. However, this paper proposes a particular QFIE-based Fuzzy Rule Based System (FRBS) for ED, which we dub as optimized QFIE, in which the structure of the circuit implementing QFIE has

* Corresponding author at: Dipartimento di Fisica, Università degli Studi di Napoli Federico II, Via Cintia, Napoli, I-80126, Italy.
E-mail address: roberto.schiattarella@unina.it (R. Schiattarella).

https://doi.org/10.1016/j.asoc.2025.113866
Received 24 March 2025; Received in revised form 29 August 2025; Accepted 2 September 2025
Available online 10 September 2025
1568-4946/

been optimized to run on NISQ computers. This structure reduces the overall width of the quantum circuit and make use of a single quantum operation to fire all the rules in the classical equivalent FRBS for ED, simplifying considerably the quantum circuit to be executed. Firstly, to assess the efficacy of this optimized implementation, a comparative analysis of the control surfaces obtained from both the standard and optimized QFIE circuits is conducted across a range of evaluation scenarios. These scenarios encompass ideal algorithm simulation, execution on noisy quantum simulators, and implementation on a NISQ device. Our findings demonstrate that the optimized quantum circuit matches the classical performance while reducing the computational overhead.

Subsequently, we utilize the proposed quantum-fuzzy ED system to conduct experiments on various images from the Berkeley Segmentation Data Set and Benchmarks 500 (BSDS-500) dataset [12], and we find that, regardless of image complexity, the performance remains robust across all evaluation scenario. These results are compared with those obtained with standard edge detection techniques such as Canny [13] and Sobel [14], showing that our approach yields superior performances in terms of Structural Similarity Index Measure (SSIM), reflecting superior image quality and structural preservation. We emphasize that our benchmarking focuses on classical, well-established detectors to provide a clear and fair baseline. While other soft computing and deep learning-based methods are undoubtedly relevant, their dependence on supervised training and parameter optimization places them beyond the scope of this initial feasibility study.

Our findings demonstrate that ED is achievable on current noisy quantum devices, marking a significant step toward quantum-enhanced image processing computed on NISQ devices.

The remaining paper is organized as follows: Section 2 analyzes the literature about ED mechanisms; Section 3 introduces the basic concepts of gradient-based methods for ED, quantum computing and QFIE; Section 4 reports the proposed approach to perform ED by using quantum fuzzy inference, as well as it introduces an optimized version of the quantum circuit implementing QFIE for this particular task; In Section 5 we present the experiments conducted to evaluate the proposed approach and its results. Additionally, we compare these results with those obtained with Canny [13] and Sobel [14] methods. Finally, we evaluate the suitability of our model for execution on NISQ devices by analyzing its resilience to the noise. Section 6 concludes the paper.

2. Related papers

In the literature, for classical computers, various approaches for ED have been developed over the years; the reader can refer to [15] for a comprehensive review.

The approaches for ED can be broadly categorized into four groups. The first group contains gradient-based methods. These methods detect the edges by identifying regions with significant intensity gradients, with well-known examples including the Sobel [14], Prewitt [16], and Roberts [17] operators, as well as the Canny edge detector [13].

The second group comprises Laplacian-based methods. These utilize the second derivative of image intensity to detect edges, with popular techniques such as the Laplacian of Gaussian (LoG) [18] and Difference of Gaussians (DoG) [19].

The third group consists of soft computing approaches, such as Genetic Algorithms and Fuzzy Logic methods. Genetic algorithms have been employed for ED, as shown in [20], where two-dimensional chromosomes and meta-level operators improve convergence. This approach demonstrated superior performance in noise robustness, convergence rate, and edge image quality compared to other techniques. Similarly, in [21], a memristive network-based genetic algorithm was introduced for ED, where image pixels act as individuals in the population, achieving improved performance over memristor-based swarm intelligence methods.

Fuzzy Logic has also been widely applied in ED, as evidenced by several key studies. In [22], the authors introduced the "fuzzy Sobel" method, which adaptively determines parameters to enhance edge clarity and addressing the limitations of traditional methods. The robustness of fuzzy approaches to impulsive noise was demonstrated in [23], achieving superior sensitivity and detail preservation. By contrast, [8–10,24] explored fuzzy ED using If-Then rules and edge continuity criteria, highlighting its robustness under high noise levels and its ability to outperform Sobel and Canny detectors without requiring manual parameter tuning. Expanding on these concepts, in [25–30], the authors developed type-2 fuzzy systems that more effectively handle uncertainty. These systems have improved ED and integrated seamlessly with hybrid approaches, such as modular neural networks. Collectively, these studies underscore the adaptability and resilience of fuzzy logic in modern ED, offering significant improvements over conventional methods.

Finally, in recent years, the fourth group, including machine learning methods, has gained significant popularity. These methods leverage algorithms to learn and identify edges from labeled training data. In [31], for example, the authors introduced a fully convolutional neural network for ED, utilizing a VGG16-based encoder [32] and a decoder with alternating anti-pooling and convolution layers for dense predictions. In [33] it was proposed an end-to-end framework for semantic ED, leveraging Residual-Network-based deep semantic learning and multi-label loss for improved edge localization and suppression of non-edge pixels. In [34], an advanced ED technique was introduced by classifying edges into four types (reflectance, illumination, normal, depth) with independent decoders for feature extraction, spatial cue integration, and attention-based refinement for precise predictions.

If the above approaches have been largely exploited on classical computers, recently, significant progress has been made in developing ED, and more in general, image processing algorithms for quantum computers.

Image processing stands to benefit significantly from the paradigm shift from classical to quantum computation. Although still an emerging field, a growing body of literature [5,35–45] and several review articles [1,46–49] have already contributed to establishing quantum image processing as a promising research direction.

Among the different methods proposed, we underline that recent advances in quantum machine learning (QML) have shown encouraging results in image classification and beyond. The field of image classification, in particular, has seen a surge of interest, with hybrid quantum–classical convolutional neural networks (CNNs) explored for both classification [50–58] and image generation [59,60].

Among the most promising approaches are hybrid quantum neural networks (HQNNs) [50], which combine classical deep learning architectures with quantum components such as parameterized quantum circuits (PQCs). These hybrid systems aim to enhance representational power while maintaining scalability through classical structures. However, it is important to note that most of these models have been evaluated primarily in simulated environments, and their effectiveness on real quantum hardware has yet to be fully demonstrated.

In [51], the authors introduce a quanvolutional layer, a quantum counterpart to classical convolutional filters, where localized transformations are applied via random quantum circuits. Experiments on the MNIST dataset suggest that quantum convolutional neural networks (QCNNs) may outperform classical CNNs in both accuracy and training efficiency.

Similarly, in [58] it is proposed a tree-structured hybrid encoding scheme that combines angle and amplitude encoding, offering flexible control over the depth and width of quantum circuits. Their quantum CNN achieves faster convergence and higher accuracy than its classical equivalent, even after dimensionality reduction. Again, these results were obtained using quantum simulators, as current hardware remains constrained in terms of qubit count and noise levels.

Among the recent quantum image processing applications, we can find various methods for quantum ED. However, most of these differ significantly from the approach presented in this work. Specifically, our method is designed to run on currently available NISQ devices, while many existing approaches remain theoretical or are tailored for future fault-tolerant quantum computers. Some of these methods, such as the quantum SUSAN ED classifier in [44] and the approach in [37], are based on quantum or quantum-inspired genetic optimization algorithms and thus fall outside the scope of optimization-free techniques. Others attempt to replicate classical gradient-based ED algorithms in the quantum setting. For example, in [41], the authors propose a quantum version of the Marr–Hildreth ED algorithm that employs quantum multipliers and adders. However, the circuit complexity of this implementation renders it infeasible for execution on NISQ hardware.

Similarly, several works – including [39,42,43] – have developed quantum versions of the Sobel operator. These approaches rely on a COPY operation and a quantum black box to calculate image gradients in parallel. Yet, to date, no efficient or scalable implementation of these operations exists for current quantum devices. The QLaplacian method presented in [40], which combines a classical Laplacian operator with the NEQR image encoding model [36], also requires a fault-tolerant quantum computer to be implemented effectively.

In [38], the authors simulate a quantum image encoding scheme where pixel values are represented in amplitude, and positions in computational basis states. For edge detection, they propose a quantum algorithm that uses only a single-qubit operation, independent of image size. While theoretically elegant, this method requires full quantum encoding of the image and an exponential number of measurements to extract edge information classically, making it currently impractical for real-world applications.

Unlike the works mentioned above, in [45], the authors present one of the first quantum edge detection (ED) methods that like our approach is executable on real NISQ devices and tested on simple real-world images. In this approach, the quantum computer is used to compute the inner product between image gradients and an edge-detection mask. For a one-dimensional mask, it requires a single-qubit circuit for each gradient direction, which can be executed sequentially or in parallel. The total depth of the transpiled circuit related to a given direction, in the *ibmq_sherbrooke* backend, is 6, resulting in an execution time of approximately 1.3 μs. However, as the dimensionality of the mask increases, the circuit complexity grows significantly. For a two-dimensional mask, three qubits are required, and the transpiled circuit reaches a depth of 58, with an estimated execution time of about 7 μs. Nevertheless, the quantum component in their framework is limited to computing inner products, with the final edge decision delegated to a classical hard thresholding function. While this marks a notable step forward, it still suffers from well-known drawbacks of classical thresholding—such as sensitivity to noise, limited adaptability, and suboptimal edge localization. In contrast, our fuzzy-based approach performs the actual edge inference on a quantum device, providing greater flexibility and robustness in decision-making and potentially improving edge quality in noisy or low-contrast scenarios.

Considering the above literature analysis, the proposed paper consists of the first application of a QFIE for image ED executed on NISQ devices. Our approach addresses the need for quantum algorithms that can perform effectively on NISQ devices. Experimental results demonstrate that our classical fuzzy inference system for ED matches the efficiency of other state-of-the-art methods. Moreover, it offers a significant advantage: its seamless adaptability to quantum devices while requiring minimal quantum resources. Notably, the results achieved using IBM's NISQ devices are comparable in quality to those obtained on classical computers.

3. Basic concepts

3.1. Gradient based methods for edge detection

Given an observed two-dimensional image I, ED can be seen as a step of the instance segmentation task, aimed at finding the curve B that approximates the boundaries of the regions/components of I. Variational approaches to ED problem require the minimization of an energy functional structured as the sum of a data fidelity term, which measures the distance (according to a defined norm) between the observed image I and the reconstructed image f, plus a regularization term, which encodes prior information on the properties of the required solution [61,62]. Let $\Omega \subset \mathbb{R}^2$ be the image domain and $B \subset \Omega$ a set of smooth boundaries, separating regions in the image I, a general formulation of the problem takes the form,

$$\min E(f,B) = \lambda \int_{\Omega} (f-I)^2 ds + \int_{\Omega - B} |\nabla f|^2 ds + \gamma |B|, \tag{1}$$

where $\lambda, \gamma > 0$ are suitable parameters and $|B|$ is the length of B [63]. This model aims at providing the estimation of both a smoothed version f of the observed image I and the boundaries B, simultaneously.

Problem (1) can be solved by deriving the corresponding Euler–Lagrange equations and applying finite-difference schemes (*optimize then discretize* approach) or, alternatively, one can first discretize the problem and then apply a suitable numerical optimization method (*discretize then optimize* approach). Under special assumptions, a solution of the variational problem can be equivalently formulated as the convolution with a specific kernel operator (see, e.g., [64]). Indeed, this ties in with different techniques that proceed by separating the task into a pre-filtering phase, where the image is preprocessed for a certain purpose (e.g., denoising/deblurring) followed by an ED phase based on the identification of discontinuities in image gradient magnitude, i.e. by finding the maximum value of the first derivative, or zero-crossing of the second derivative. These types of methods have gained popularity for their ease of implementation and computational efficiency; they rely on the observation that pixel intensity shows a sudden change at the boundary between objects. In particular, given an image I, the first partial derivatives along the two directions x and y can be computed by finite difference approximations at one-pixel intervals. For example, using backward difference, for any pixel (x, y), we have

$$I_x := \frac{\partial I(x,y)}{\partial x} \approx I(x-1,y) - I(x,y), \tag{2}$$

$$I_y := \frac{\partial I(x,y)}{\partial y} \approx I(x,y-1) - I(x,y). \tag{3}$$

The results of these operators are equivalent to the convolution of the image by suitable kernels. Indeed, given two discrete functions represented, respectively, by a matrix $A \in \mathbb{R}^{n\times m}$ (the kernel) and a matrix $B \in \mathbb{R}^{k\times p}$, the convolution is the $(n+k-1)\times(m+p-1)$ matrix defined component-wise as

$$[A * B]_{ij} = \sum_s \sum_t A_{st} B_{i-s+1,j-t+1}, \tag{4}$$

where $s = \max(1, i-n+1)\ldots\min(i,m)$ and $t = \max(1, j-k+1)\ldots\min(j,p)$. Thus, the expressions (2)–(3) correspond to the convolution of the image by vectors $G_x = [-1, 1]$ and $G_y = [-1, 1]^T$, respectively, i.e. the gradient of the image can be approximated pixel-wise as

$$\nabla I(x,y) \approx \begin{bmatrix} G_x * I \\ G_y * I \end{bmatrix}, \tag{5}$$

with prefixed boundary conditions.

Hence, an approximation of the gradient magnitude at pixel (x, y) is given by

$$\|\nabla I(x,y)\|_2 = \sqrt{G_x^2 + G_y^2}, \tag{6}$$

and the corresponding direction can be estimated by the angle formed with the x axis

$$\theta = \arctan \frac{G_y}{G_x}, \tag{7}$$

which corresponds to the direction of maximum change of pixel intensity, therefore, the magnitude of the gradient provides the amount of this change. Well-known ED techniques [13,14,16,17] arise from variants of the previous scheme, by computing more accurate discrete approximations of the first-order partial derivatives along two perpendicular directions, considering $n \times n$ convolutional kernels with $n \geq 2$ (small), and then detecting the edges by thresholding the gradient magnitude.

3.2. Quantum computing

Quantum Computing is an innovative computational paradigm that exploits qubits as basic information units. Differently from a classical bit, a qubit is a two-level quantum state that can be expressed as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{8}$$

where $\alpha, \beta \in C$ and $|\alpha|^2 + |\beta|^2 = 1$. Physically, once a qubit is measured a classical bit is retrieved, and the stochastic result will be 0 or 1 with a probability of $|\alpha|^2$ and $|\beta|^2$ respectively. A quantum state describing a system composed of n qubits is given by the tensor product of single-qubit states and it can be expressed as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle\,, \tag{9}$$

where i is the integer representation of the binary bit-string representing one of the 2^n basis states. The qubits are manipulated through unitary operators, which are physically implemented as quantum gates. Therefore, a quantum algorithm can be formulated as a quantum circuit composed of a sequence of quantum gates, transforming an input quantum state into an output quantum state. From the latter, classical output information can be retrieved by executing the quantum circuit several times and measuring the qubits of interest. The number of repetitions is usually named as the number of shots.

By exploiting the peculiarities of quantum superposition, entanglement, and interference, many quantum algorithms that outperform the capabilities of their classical counterparts have been proposed. For instance, very well-known representative examples are Shor's algorithm [65] and Grover's algorithm [66]. The former is an efficient quantum algorithm for integer factoring. This problem does not have a classical solver that can factorize an integer n encoded with b bits in polynomial time, while Shor's algorithm achieves the goal in $O(b^3 \log b)$ time, using $O(b^2 \log n \log\log n)$ quantum gates. On the other hand, Grover's quantum algorithm searches for one marked entry in an unstructured, unordered list of N entries, using only $O(\sqrt{N})$ queries instead of the $O(N)$ queries required classically. However, for the practical usage of these algorithms, a fault-tolerant quantum computer equipped with thousands of qubits would be required. Conversely, although the development of quantum hardware has been impressive in the last decade, today's quantum hardware is still characterized by a limited number of qubits (a hundred or so) that are extremely noisy. These devices are known as NISQ computers [6].

To model quantum noise, one usually uses the density matrix formalism, which generalizes pure states to also include *mixed states*—statistical mixtures of pure states affected by decoherence or noise. The density matrix is defined as

$$\rho = \sum_j p_j \left|\psi_j\right\rangle \left\langle\psi_j\right|, \tag{10}$$

and for a pure state $|\psi\rangle$, reduces to

$$\rho = |\psi\rangle\langle\psi|\,. \tag{11}$$

More generally, noisy processes are modeled by quantum channels using Kraus operators, namely,

$$\mathcal{N}(\rho) = \sum_i K_i \rho K_i^{\dagger}. \tag{12}$$

Many types of errors can occur, like the *readout error*, that flips the outcome with probability r,

$$\mathcal{N}_{\rm ro}(\rho) = (1-r)\rho + r\,X\rho X, \tag{13}$$

or the *Depolarizing channel*, where the state ρ is replaced by a uniform mixture of the three Pauli errors with total weight s,

$$\mathcal{N}_{\rm dep}(\rho) = (1-s)\,\rho + \frac{s}{3}\big(X\,\rho\,X + Y\,\rho\,Y + Z\,\rho\,Z\big). \tag{14}$$

Other types of error are, for example, bit flip error, phase flip error, and amplitude damping. Their Kraus decomposition can be read in [67].

These models are essential to understand and mitigate noise in NISQ devices. Since noise is non-negligible in such systems, it can significantly affect computational outcomes. Consequently, finding actual quantum advantages exploiting quantum algorithms suitable for being executed on NISQ devices is the current most important topic in quantum computing. The proposed paper moves an important step in this direction, by designing an efficient fuzzy system for performing ED on images, whose engine is implemented as a quantum algorithm that can be executed even on NISQ devices. The next section highlights the peculiarities of this quantum fuzzy inference engine.

3.3. The quantum Fuzzy inference engine

This section aims to summarize the principal aspects of the QFIE introduced in [11]. This algorithm exploits an amplitude encoding of fuzzified values in quantum states and a formalization of the rule base of an FRBS as a Boolean quantum oracle that can be implemented as a set of gates on a quantum computer. To detail these two peculiarities, let us consider a fuzzy system composed of n input variables $x_1, x_2, \dots, x_n$ with m fuzzy sets defined over the related universe of the discourse and one output variable y with m_y fuzzy sets. Also, let us denote the fuzzy sets defined for the ith input variable as $\mathcal{S}_1^i, \mathcal{S}_2^i, \dots \mathcal{S}_m^i$ and the fuzzy sets for the output variable as $\mathcal{B}_1, \mathcal{B}_2, \dots \mathcal{B}_m$. Finally, let us suppose that the fuzzy system has a complete set of rules expressed as

$$\text{IF } (x_1 \text{ is } \mathcal{S}_j^1) \text{ and } (x_2 \text{ is } \mathcal{S}_l^2) \text{ and } (x_n \text{ is } \mathcal{S}_k^n) \text{ THEN } y \text{ is } \mathcal{B}_r, \tag{15}$$

where j, l, n can be any integer between 1 and m, whereas r is an integer between 1 and m_y.

Fig. 1 reports the workflow of a fuzzy system equipped with the quantum fuzzy inference engine. According to such a workflow, the first step in a fuzzy system equipped with QFIE consists of performing classical fuzzification. Therefore, denoting with X_i the crisp input value for the variable x_i, it is computed the membership degrees $\mu_1^i(X_i), \mu_2^i(X_i), \dots, \mu_m^i(X_i)$ of X_i to each fuzzy sets $\mathcal{S}_1^i, \mathcal{S}_2^i, \dots \mathcal{S}_m^i$. For the sake of simplicity in the mathematical notation it will be used hereafter to describe QFIE, let us suppose that for each fuzzy variable, the related fuzzy sets form an exact fuzzy partition, i.e.

$$\sum_{j=1}^{m} \mu_j^i(X_i) = 1 \quad \forall i \in [1, n]. \tag{16}$$

At this point, the quantum circuit implementing QFIE consists of $n+1$ quantum registers. The first n registers contain the information on the fuzzified input values and each of them is composed of $\eta = \lceil \log_2(m) \rceil$ qubits, whereas the remaining register is composed of m_y qubits. Each input register is initialized as

$$|\psi_i\rangle = \sum_{j=1}^{\eta} \sqrt{\mu_j^i(X_i)}\left|\mathcal{S}_j^i\right\rangle, \tag{17}$$

where with abuse of notation, the bitstring representing the basis state whose amplitude is $\sqrt{\mu_j^i(X_i)}$ is represented with the same notation of the fuzzy set $\mathcal{S}_j^i$. The initialization in Eq. (17) is possible if the condition in Eq. (16) holds. However, as discussed in [11] this encoding strategy is possible even if it is true the lighter condition $\sum_{j=1}^{m} \mu_j^i(X_i) \leq 1$. In that case, it is required to add a qubit to the ith quantum register and

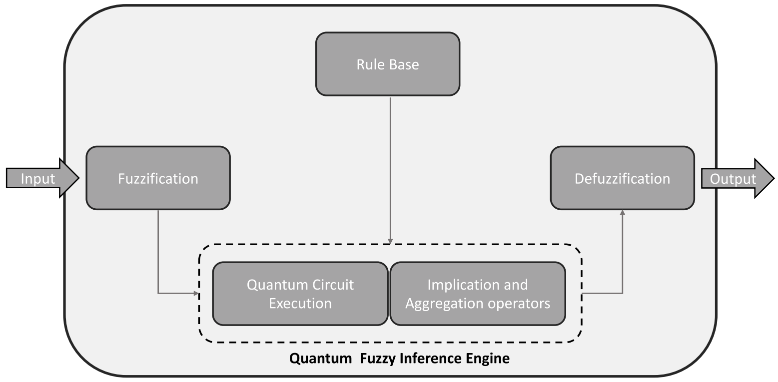


Fig. 1. Workflow of a fuzzy system equipped with the QFIE.

Fig. 2. Swan photo from BSDS-500 dataset [12] in RGB (a) and grayscale (b) color models.

encode the quantity $\sqrt{1-\sum_{j=1}^{m}\mu_j^i(X_i)}$ as the amplitude of a basis state that does not correspond to any fuzzy set.

By repeating the initialization in Eq. (17) for all the input registers, the overall state describing the n registers will be expressed as

$$|\psi\rangle = \sum_{j=0}^{\eta-1}\sum_{l=0}^{\eta-1}\cdots\sum_{k=0}^{\eta-1}\sqrt{\mu_j(X_1)\mu_l(X_2)\ldots\mu_k(X_n)}\left|S_j^1, S_l^2, \ldots, S_k^m\right\rangle. \tag{18}$$

In other words, thanks to the tensor product between the states initialized as in Eq. (17), the resulting quantum state is a state having encoded in each amplitude the fire strength of a different possible antecedent of the rules expressed as in Eq. (15) computed by using the product t-norm. As suggested in [11], by denoting with $\mathcal{A}_s$ the set of all possible antecedents, the state in Eq. (18) can be rewritten as

$$|\psi\rangle = \sum_{a\in\mathcal{A}_s}\sqrt{F_a}\,|a\rangle. \tag{19}$$

At this point, a quantum oracle is used to create entanglement between the antecedents and the related consequents. In detail, each consequent fuzzy set is represented in the output quantum register as the basis state having all zeros and just one qubit in the state $|1\rangle$. Explicitly, the output fuzzy set $\mathcal{B}_1$ is represented by the state $|100\ldots0\rangle$, the output fuzzy set $\mathcal{B}_2$ by the state $|010\ldots0\rangle$, etc... Considering this, a quantum oracle O_f is applied to the quantum circuit and it results in

$$|\psi_Y\rangle = \mathcal{O}_f\,|\psi\rangle\,|\bar{0}\rangle = \sum_{i=1}^{m_Y}\sum_{a\in\mathcal{A}_s^i}\sqrt{F_a}\,|a\rangle\,|\mathcal{B}_i\rangle, \tag{20}$$

where $\mathcal{A}_s^i$ is used to denote the subset of antecedents in $\mathcal{A}_s$ that have as consequent in the related rule the output fuzzy set $\mathcal{B}_i$. This quantum circuit is executed for a given number of shots $\mathcal{N}$, in order to retrieve the probability of measuring each bitstring $\mathcal{B}_i$ in the output register. From Eq. (20) it is easy to see that, by measuring the output register, the probability of getting $|\mathcal{B}_i\rangle$ is given by

$$P_{\mathcal{B}_i} = \sum_{a\in\mathcal{A}_s^i} F_a \quad \forall i\in[1, m_Y]. \tag{21}$$

Subsequently, the set of probabilities $\{P_{\mathcal{B}_i}\}_{i=1}^{m_Y}$ is used by an implication operator which performs the following action:

$$\mu_i^y = \min(P_{\mathcal{B}_i}, \mu_{\mathcal{B}_i}(y)), \tag{22}$$

where with $\mu_{\mathcal{B}_i}(y)$ it is denoted the membership function describing the output fuzzy set $\mathcal{B}_i$. Finally, the aggregation operator combines all the cut output membership functions μ_i^y by using the maximum co-norm operator. This results in an output fuzzy set that can be classically defuzzified to obtain the crisp output value. Commonly used defuzzification approaches are the centroid defuzzification or the Largest of Maximum (LOM) [68].

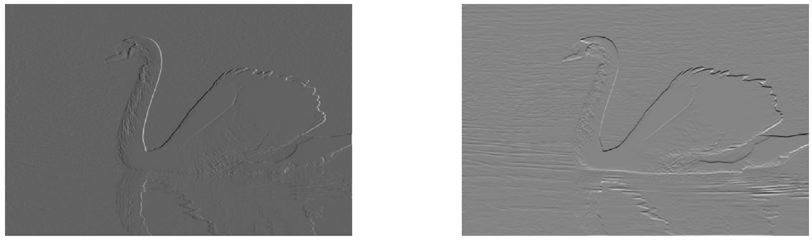
According to the analysis carried out in [11], in terms of oracle queries the computational complexity for QFIE is $O(n)$, whereas the computational complexity of an equivalent classical fuzzy system is $O(m^n)$.

4. Quantum Fuzzy logic for the edge detection problem

This section aims to introduce the proposed QFIE-based FRBS for ED. Specifically, it first presents the classical equivalent FRBS for this task, then, explores how the QFIE, as introduced in [11], can be exploited to develop the quantum engine of such an FRBS. Finally, it discusses how this quantum algorithm can be strongly optimized, obtaining a much shallower equivalent quantum circuit, optimal for execution on NISQ devices.

4.1. Classical FRBS for edge detection

As described in [69], gradient-based methods (Section 3.1) can be effectively combined with FRBSs for robust edge detection. These approaches overcome the limitations of the hard thresholding techniques in the presence of noise in the images. To detail how FRBS can be used to infer edges, let us consider a given image that is converted from its RGB color model to grayscale, with gray levels ranging from 0 (black) to 1 (white). An example is reported in Fig. 2. On this grayscale



a. Gradient along X-axis b. Gradient along Y-axis

Fig. 3. Gradient along x (a) and y (b) axis computed on the grayscale version of the swan photo.

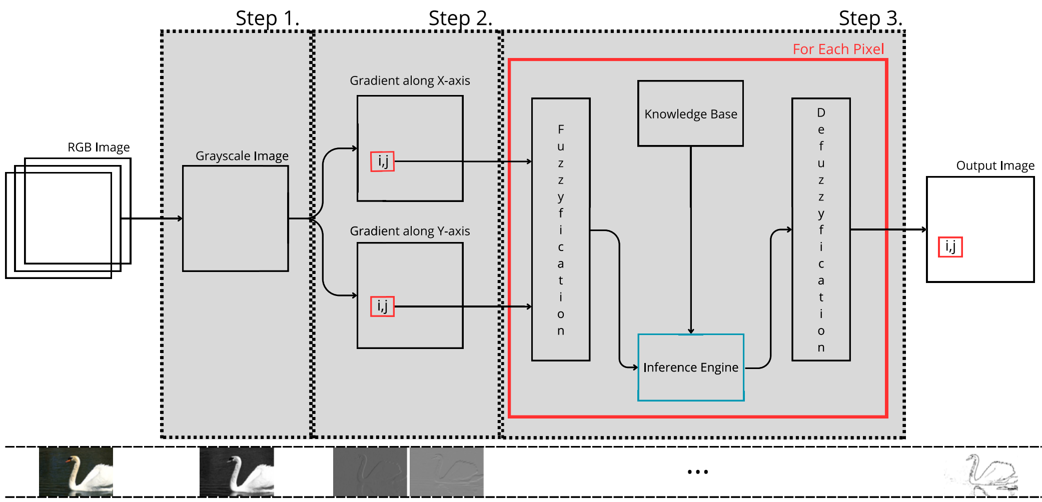


Fig. 4. Workflow of FRBS-based edge detection process.

image, it is possible to compute the axis gradients I_x and I_y according to Eq. (5). The result of the two gradient operations is reported in Fig. 3. By using these quantities, it is possible to discriminate a pixel that lies in a uniform region from a pixel on an edge. Indeed, in the former scenario, the gradient of the pixel will be close to zero in both directions, whereas in the latter case at least in one direction there will be a non-zero gradient. The FRBS has the goal of highlighting the pixels on the edges, generating a new image where these pixels are represented as black, while the pixels belonging to uniform regions are depicted as white. Unlike a simple threshold-based method, which may struggle with noise and weak edges due to its rigid decision boundaries, FRBS provides a more adaptive approach by incorporating gradual transitions between edge and non-edge pixels. This flexibility allows for better edge preservation and robustness in varying image conditions. The whole image representing the edges is obtained by running this inference process on all the $n \times n$ pixels of the input image. Fig. 4 graphically summarizes the aforementioned workflow.

According to [69], a FRBS suitable to perform this inference process is composed of two input variables I_x and I_y ranging from -1 to 1 and the output pixel intensity (P_I) variable ranging from 0 to 1. For the former, two fuzzy sets named *one* and *zero* are defined, or, equivalently, *one* and NOT *one*. The membership functions for these fuzzy sets are Gaussian functions of the form

$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}}, \tag{23}$$

with $c = 0$ and $\sigma = 0.1$, and are reported in Fig. 5a.

Note that for both the input variables, exact fuzzy partitions are defined, i.e., the sum of the two membership degrees is always one,

$$\mu_0(x) + \mu_1(x) = 1 \quad \forall\ x \in [-1, 1], \tag{24}$$

where by x is denoted the generic value of I_x and I_y, while $\mu_0(x)$ and $\mu_1(x)$ represent the membership degree of x to the fuzzy sets *zero* and *one*, respectively. Similarly, for P_I two fuzzy sets are considered, namely *black* and *white*. These are modeled respectively through L-trapezoidal, defined as

$$\mu_B(x) = \begin{cases} 1, & x \le a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \ge b, \end{cases} \tag{25}$$

with $a = 0.0$ and $b = 0.7$, and R-trapezoidal membership functions, defined as

$$\mu_W(x) = \begin{cases} 0, & x \le a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \ge b, \end{cases} \tag{26}$$

with $a = 0.1$ and $b = 1.0$. These are reported in Fig. 5b.

The set of If-Then rules representing the rule base of this FRBS is composed of two rules, namely:

$$\begin{aligned} &\text{if } I_x \text{ is zero AND } I_y \text{ is zero THEN } P_I \text{ is white} \\ &\text{if } I_x \text{ is one OR } I_y \text{ is one THEN } P_I \text{ is black} \end{aligned} \tag{27}$$

The next section details how this FRBS can be adapted to be equipped with the QFIE introduced in Section 3.3 to perform the fuzzy inference process on quantum computers.

4.2. QFIE-based implementation of FRBS for edge detection

This section discusses the integration of the QFIE, introduced in Section 3.3, into the fuzzy-based ED mechanism illustrated in Fig. 4. Specifically, the QFIE replaces the classical inference engine in the FRBS, enabling the execution of fuzzy rules evaluation on a quantum computer. However, the original QFIE proposed in [11] is capable of firing fuzzy rules where the antecedent part is composed solely of AND connections between the different clauses. Consequently, to perform

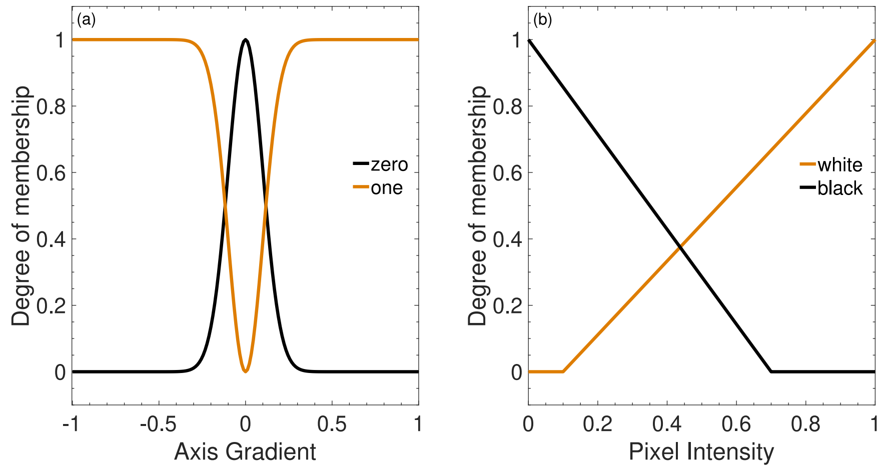


Fig. 5. Input (a) and output (b) membership functions.

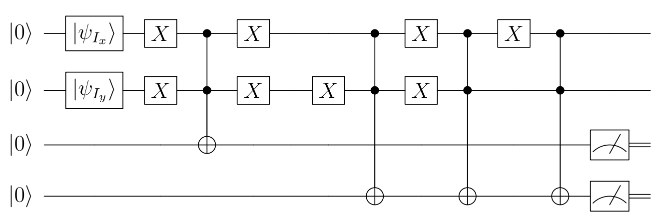


Fig. 6. QFIE quantum circuit.

the inference process of the aforementioned FRBS through QFIE, it is necessary to modify the original rule set described in (27). The equivalent set of fuzzy rules is reported in (28).

$$\begin{array}{l}\text{if } I_x \text{ is zero AND } I_y \text{ is zero THEN } P_I \text{ is white}\\ \text{if } I_x \text{ is one AND } I_y \text{ is zero THEN } P_I \text{ is black}\\ \text{if } I_x \text{ is zero AND } I_y \text{ is one THEN } P_I \text{ is black}\\ \text{if } I_x \text{ is one AND } I_y \text{ is one THEN } P_I \text{ is black}\end{array} \tag{28}$$

By considering this set of rules, and the membership functions reported in Fig. 5 it is now possible to analyze the quantum circuit implementing the related QFIE. The condition of an exact fuzzy partition for both the input variables is true (see Eq. (24)). In this case, just one qubit for each input variable is used to encode the fuzzified values as described in Eq. (17). On the other hand, according to the original QFIE algorithm, the output register is composed of a number of qubits equal to the number of fuzzy sets defined in the output variable universe of discourse, which in the above FRBS is two. Overall, the quantum circuit implementing QFIE for this FRBS is composed of $\eta = 4$ qubits. Finally, according to [11], the quantum oracle in Eq. (20) is implemented through L multi-controlled not gates, where L is equal to the number of fuzzy rules in the rule-base. Fig. 6 reports the whole quantum circuit implementing the QFIE for the FRBS for ED.

Formally, let us consider two input values I_x and I_y for the gradients computed for the pixel (x, y) of a given input gray image. The input registers of the quantum circuit implementing QFIE would be initialized by exploiting the method introduced in [70] as follows

$$\begin{aligned}\left|\psi_{I_x}\right\rangle &= \sqrt{\mu_0(I_x)}\,|0\rangle + \sqrt{\mu_1(I_x)}\,|1\rangle\,,\\ \left|\psi_{I_y}\right\rangle &= \sqrt{\mu_0(I_y)}\,|0\rangle + \sqrt{\mu_1(I_y)}\,|1\rangle\,.\end{aligned} \tag{29}$$

Therefore, the overall input state of the circuit can be explicitly defined as

$$\begin{aligned}|\psi_I\rangle = &\sqrt{\mu_0(I_x)\mu_0(I_y)}\,|00\rangle + \sqrt{\mu_0(I_x)\mu_1(I_y)}\,|01\rangle +\\ &+\sqrt{\mu_1(I_x)\mu_0(I_y)}\,|10\rangle + \sqrt{\mu_1(I_x)\mu_1(I_y)}\,|11\rangle\,.\end{aligned} \tag{30}$$

After the application of the quantum oracle on this input state and the output state initialized in $|00\rangle$, the state describing the whole set of qubits will be

$$\begin{aligned}O_f\,|\psi_I, 00\rangle = &\sqrt{\mu_0(I_x)\mu_0(I_y)}\,|00,10\rangle + \sqrt{\mu_0(I_x)\mu_1(I_y)}\,|01,01\rangle +\\ &+\sqrt{\mu_1(I_x)\mu_0(I_y)}\,|10,01\rangle + \sqrt{\mu_1(I_x)\mu_1(I_y)}\,|11,01\rangle\,.\end{aligned} \tag{31}$$

Finally, the two output qubits are measured, and the following two probabilities can be retrieved

$$\begin{aligned}P_w &\equiv P(|10\rangle) = \mu_0(I_x)\mu_0(I_y),\\ P_b &\equiv P(|01\rangle) = \mu_0(I_x)\mu_1(I_y) + \mu_1(I_x)\mu_0(I_y) + \mu_1(I_x)\mu_1(I_y).\end{aligned} \tag{32}$$

The probabilities P_w and P_b are then used to perform the implication step as in Eq. (22). Finally, after the aggregation, the output fuzzy set can be classically defuzzified to obtain the pixel intensity of the new image.

4.3. Optimized QFIE for edge detection

This section details how it is possible to modify the QFIE quantum circuit reported in Fig. 6 to obtain an equivalent and more efficient quantum circuit both in terms of depth and number of qubits. This new version of QFIE requires one qubit for each input variable and one qubit also for encoding the information on the output variable. This reduces the overall width of the quantum circuit to $\eta_{new} = 3$ qubits. As for the standard QFIE, the first step is the initialization of the two quantum states related to the two input variables. These are initialized as in Eq. (29), but not using the initialization approach proposed in [11] based on the methodology introduced in [70]. Indeed, it is possible to obtain the quantum states in Eq. (29) just by using an R_y gate with a properly defined angle θ. Considering that $R_y(\theta)$ is defined as

$$R_y(\theta) = \begin{pmatrix}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{pmatrix}, \tag{33}$$

the states in Eq. (29) can be obtained by setting the two relative angles as

$$\theta_x = 2\arccos\left(\sqrt{\mu_0(I_x)}\right); \quad \theta_y = 2\arccos\left(\sqrt{\mu_0(I_y)}\right). \tag{34}$$

Secondly, the structure of the quantum oracle O_f can be reduced to an equivalent quantum oracle O_f^n composed of a single Toffoli gate, reducing the whole quantum circuit to the one reported in Fig. 7.

Indeed, the application of this quantum oracle to the state reported in Eq. (30) and on the output qubit initialized in $|0\rangle$, results in the state

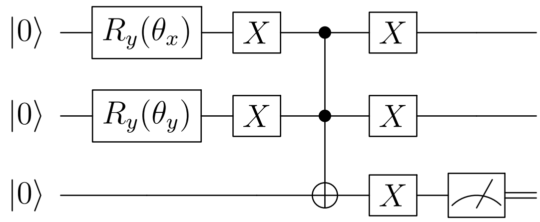


Fig. 7. Optimized quantum circuit.

Table 1
Comparison between the standard and optimized QFIE circuit features before and after transpilation on the *ibmq_sherbrooke* backend.

Feature	Standard QFIE		Optimized QFIE	
	Original	Transpiled	Original	Transpiled
Number of qubits	4	4	3	3
Circuit depth	11	198	5	43
One-Qubit gates		243		52
Two-Qubit gates		44		9
Execution time		27.98 μs		4.45 μs

$$O_f^n\,|\psi_I, 0\rangle = \sqrt{\mu_0(I_x)\mu_0(I_y)}\,|00,0\rangle + \sqrt{\mu_0(I_x)\mu_1(I_y)}\,|01,1\rangle + \sqrt{\mu_1(I_x)\mu_0(I_y)}\,|10,1\rangle + \sqrt{\mu_1(I_x)\mu_1(I_y)}\,|11,1\rangle\,. \tag{35}$$

At this point, the probability of measuring the output qubit in the state $|0\rangle$ and in the state $|1\rangle$ are respectively equal to the probabilities P_w and P_b defined in Eq. (32). Therefore, the same implication and aggregation steps as for the standard QFIE can be performed to obtain an equivalent output fuzzy set, from which the pixel intensity can be retrieved by means of a classical defuzzification approach.

To summarize, the proposed optimized version of the QFIE quantum circuit for fuzzy inference in ED requires fewer qubits and a reduced number of gates compared to the standard QFIE implementation. Notably, the number of multi-controlled NOT gates reduces from four in the standard version to just one. More importantly, this optimized circuit exemplifies an early tangible advantage of the quantum approach over its classical counterpart. Even in the standard QFIE, a single oracle query suffices to complete the inference process, whereas a classical FRBS requires as many oracle calls as there are rules in the rule base. However, the construction of the quantum oracle in the standard QFIE still involves a number of multi-CNOT gates that scale linearly with the number of rules. In contrast, the proposed optimized version requires only one multi-CNOT gate to implement an equivalent oracle, whereas in the classical case, two rules must be executed.

4.4. Performing edge detection on NISQ devices

In Sections 4.2 and 4.3, it was shown how it is possible to implement the quantum circuit of QFIE to replace the classical inference engine of FRBS introduced in Section 4.1. Both the standard implementation of QFIE (Fig. 6) and its optimized version (Fig. 7) result in quantum circuits that are small enough to be executed on current NISQ hardware. However, two important clarifications need to be made. Firstly, each Toffoli gate in the quantum oracles must be transpiled according to the backend on which the operator must be executed. Indeed, on current superconductive quantum devices, it is possible to execute only operations involving one or two qubits simultaneously. The transpiled quantum circuit is therefore much more complex than the original one, and for this reason it is very sensitive to quantum noise. As an example, Fig. 8 reports the transpiled quantum circuit on the *ibmq_sherbrooke* quantum processor of a single Toffoli gate having the first two qubits as control and the latter one as the target. On this basis, the optimized QFIE offers a significant reduction in quantum resource usage compared to its standard counterpart. In fact, as reported in Table 1, using the *ibmq_sherbrooke* quantum processor as a reference, the transpiled optimized circuit achieves a 78% reduction in circuit depth (from 198 to 43) and an 80% reduction in the number of two-qubit gates (from 44 to 9) compared to the unoptimized version. As a consequence, the execution time, evaluated using the intrinsic function *schedule* of qiskit, gives 27.98 μs for the standard QFIE circuit and 4.45 μs for the optimized circuit. For reference, the *ibmq_sherbrooke* device has a median relaxation time (T_1) of approximately 261 μs and a median dephasing time (T_2) of approximately 210 μs. Therefore, both the standard and optimized circuits execute well within these coherence times, and the significant reduction in execution time for the optimized circuit provides increased robustness against decoherence effects.

It is worth noting that these execution times, obtained using Qiskit's intrinsic *schedule* function, reflect the pulse-level duration of the circuits. The actual end-to-end execution time on the *ibmq_sherbrooke* backend, which includes circuit loading, calibration sequences, multiple repetitions, and qubit reset, is considerably longer. For both the standard and optimized circuits, this total execution time is approximately 1 s. Nevertheless, the substantial reduction in circuit depth and gate count for the optimized QFIE still provides clear advantages in terms of error accumulation and robustness, despite the similar end-to-end durations.

This consideration underscores the significance of optimizing the QFIE quantum circuit, as introduced in Section 4.3, for ensuring the reliable execution of this algorithm on NISQ devices.

Secondly, to perform ED by exploiting the proposed QFIE-based FRBS on an input image with resolution $n \times n$, it would be required to execute the quantum circuit implementing QFIE n^2 times, i.e. one execution for each pixel of the image. However, given the challenges associated with accessing current NISQ devices due to their extensive queue times and limited computational resources, this approach is impractical for ED on real-world images. To address this challenge, this paper proposes a workaround. Specifically, the FRBSs can be executed by receiving input from every possible gradient value pair, ranging from -1 to 1, with a given step, and obtaining the related P_i. These results are stored in a lookup table so that, given an image, the edges are then detected by mapping each pixel's gradient pair to the closest gradient pair in the lookup table and obtaining its intensity value in grayscale. Considering this, the proposed QFIE-based inference mechanism for ED can be described as a hybrid approach, where the construction of the lookup table is carried out by using the quantum algorithm, while the inference from the lookup table is executed classically.

It is important to emphasize that the adoption of the lookup table is a pragmatic workaround necessitated solely by the current limitations in the accessibility of quantum hardware. As quantum computing technologies continue to advance, increasing the accessibility to these devices, the direct application of the QFIE for real-time inference will become increasingly feasible, thereby rendering the lookup table unnecessary.

5. Experiments and results

This section presents the experiments conducted to evaluate the effectiveness of the optimized QFIE-based FRBS for ED. The experiments consist of three main steps:

1. First, a comparison is made between the standard QFIE-based FRBS and the optimized QFIE-based FRBS. This evaluation, detailed in Section 5.1, focuses on analyzing the lookup tables generated by both methods and their feasibility for execution on NISQ devices.

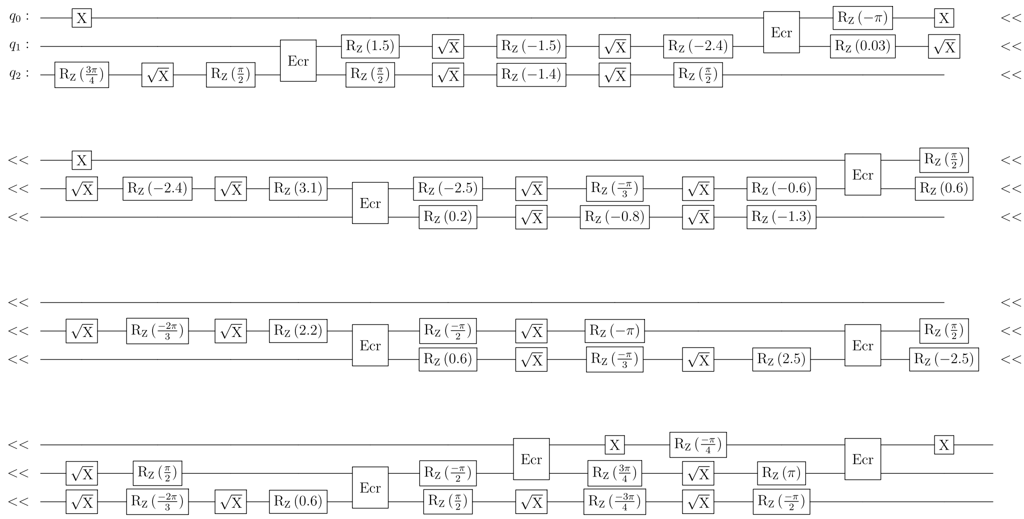


Fig. 8. Toffoli decomposition on *ibmq_sherbrooke*.

2. The second experiment, detailed in Section 5.2, involves a comparison of the optimized QFIE-based FRBS with a classical FRBS. This is achieved by applying both methods to ED on images from the BSDS-500 dataset [12]. This dataset comprises 500 full-color (RGB) natural images, in JPG format, with typical resolutions around 481 × 321 pixels, available in both horizontal and vertical orientations. It features diverse scenes and lighting conditions, making it a valuable benchmark for edge detection and other image processing tasks as testified by the large number of methods using it [71–77]. Additionally, we have verified that the statistical properties of the BSDS-500 dataset are fully consistent with those of other widely used benchmarks, such as COCO [78] and VOC2012 [79]. In particular, the gradients and grayscale distributions across the three datasets confirm that BSDS-500 is a representative benchmark for natural images. To assess the quality of detection with respect to other well-known methods, the subsection includes a comparison of both QFIE-based FRBS and classical FRBS with Canny [13] and Sobel [14] filters.
3. Lastly, in Section 5.3, an analysis was conducted to assess the quantum noise resilience of the optimized QFIE circuit. This evaluation was carried out through simulations under various noise models, reflecting realistic conditions encountered on current NISQ devices, in order to investigate the robustness of the proposed approach under different levels of quantum noise.

5.1. Optimized versus standard QFIE implementation

This section aims to compare the lookup tables described in Section 4.4 obtained when using the quantum-based FRBS equipped with the standard or the optimized version of QFIE to the one obtained by using the classical FRBS. In detail, the quantum algorithms are executed under three conditions:

- **Noiseless Simulations:** Ideal conditions without quantum noise.
- **Noisy Simulations:** Simulated quantum circuits incorporating realistic noise models.
- **Real Hardware Execution:** Implementation on actual NISQ devices.

The lookup tables are analyzed through their graphical representation as control surfaces, i.e. 3D plots where each input point (x, y) corresponds to gradient pairs (I_x, I_y), and the z-axis represents the output pixel intensity P_I. In all experiments, the gradient step is set to 0.1, resulting in lookup tables with 21 × 21 entries. The significance of this step size in ED is further discussed in Section 5.2.3. To numerically compare the different control surfaces, two metrics will be considered [80]. Say S and S' are two control surfaces to compare, the first metric considered is the *root mean square deviation (RMSD)*

$$\mathrm{RMSD} = \sqrt{\sum_{i=1}^{N} \frac{1}{N}(\mathbf{r}_i - \mathbf{r}_i')^2}, \tag{36}$$

where $\mathbf{r}_i$ and $\mathbf{r}_i'$ represent the three-dimensional coordinates of the ith point on surfaces S and S', respectively, and N is the total number of surface points.

The second metric is the *Hausdorff distance (HD)*

$$\mathrm{HD} = \max\left\{\sup_{\mathbf{r}\in S} d(\mathbf{r}, S'),\ \sup_{\mathbf{r}'\in S'} d(\mathbf{r}', S)\right\}, \tag{37}$$

where $d(\mathbf{r}, S')$ denotes the Euclidean distance between a point $\mathbf{r}$ on S and the surface S', and vice versa. Both metrics approach zero in the limit where the two surfaces are identical.

5.1.1. Experimental setup

This section reports all the details required to replicate the experiments presented in Section 5.1.2. To ensure a fair comparison between the three FRBSs considered, all of them have been equipped with a LOM defuzzification approach. The experiments carried out with the classical FRBS were performed by exploiting MATLAB, whereas the experiments involving the two versions of QFIE were performed by using the Qiskit SDK. In detail, the noisy simulations of both the standard and the optimized QFIE quantum circuit were performed by exploiting the *Fake Sherbrooke* backend, i.e. a quantum simulator that mimics the behavior of *ibmq_sherbrooke* quantum processor. In the end, the experiments carried out on real quantum hardware were computed by exploiting the *ibmq_sherbrooke* device, a 127-qubit device from IBM Quantum. The number of quantum circuit shots set for both the simulations and the execution on real hardware was 1000. Finally, the stability of the results with respect to this quantity is also discussed at the end of the section.

5.1.2. Experimental results

Fig. 9a reports the control surface obtained by using the classical FRBS to compute the output pixel intensity on the grid of gradient values ranging from −1 to 1 with steps of 0.1. This control surface reflects the fuzzy rule-set (27) and the membership functions in Fig. 5: pixel intensity (z-axis) has a spike around 0 gradient (x/y plane) and is 0 everywhere else. Similarly, the control surface in Fig. 9b has been obtained by running, respectively, the standard QFIE circuit

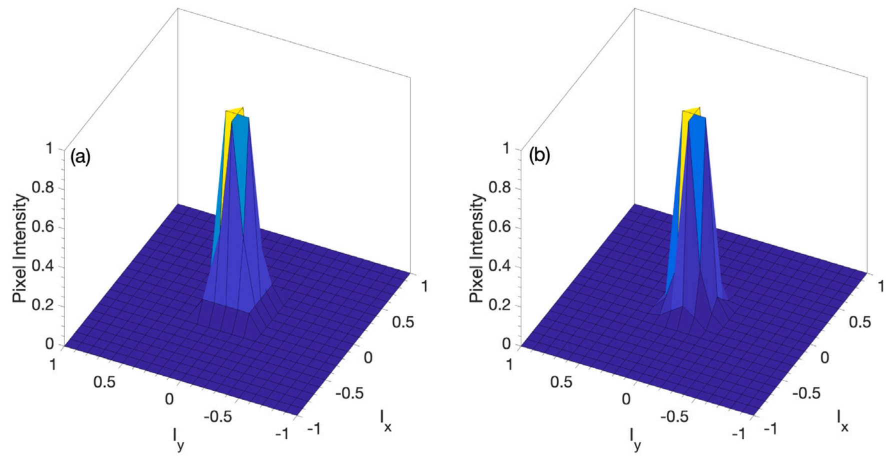


Fig. 9. Control surfaces computed on classic hardware (a) and on an ideal quantum simulator for the standard and the optimized QFIE (b).

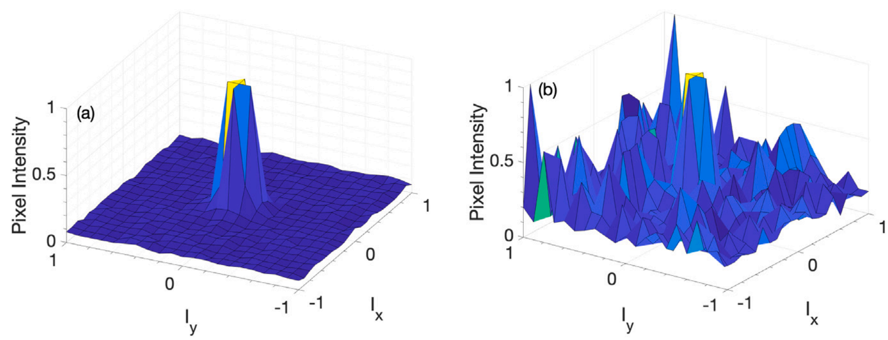


Fig. 10. Control surfaces on the noisy quantum simulator (a) and on the NISQ device (b) for the standard QFIE.

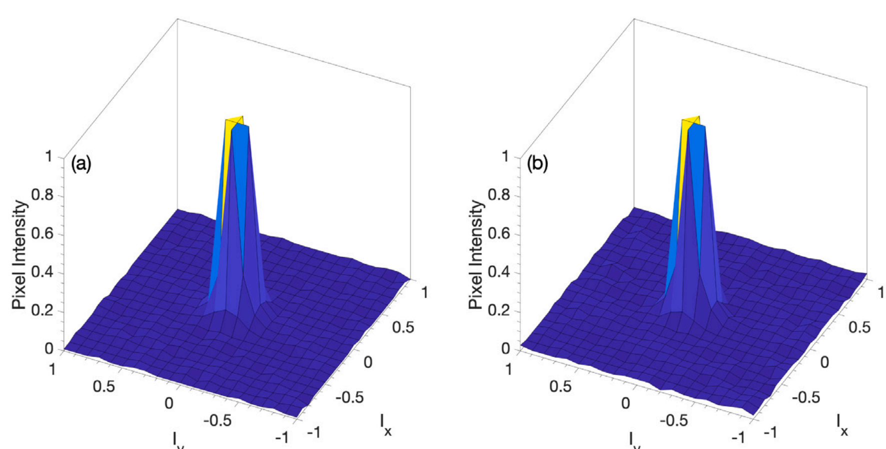


Fig. 11. Control surfaces on the noisy quantum simulator (a) and on the NISQ device (b) for the optimized QFIE.

and the optimized QFIE circuit on an ideal quantum simulator. Since Section 4.3 demonstrates the equivalence of these two quantum circuit implementations in the absence of noise, they produce identical results. Notably, the obtained surface closely aligns with the classical one.

Subsequently, Fig. 10a and b report the control surface obtained by using the FRBS equipped with the standard QFIE, under noisy simulation and real hardware execution, respectively. A similar process was followed for the optimized QFIE, producing the control surfaces in Fig. 11a and b.

Both the surface obtained from the noisy simulation of the standard and the optimized QFIE present a structure similar to the results obtained on the ideal simulator (Fig. 9b), with small deviations generated by the introduction of the noise in the simulation. The effect of these fluctuations is in general more evident in the zone with pixel intensity around zero, while no appreciable modifications appear at first glance to pixels with intensity close to one. However, the situation changes drastically when considering the experiments performed on the actual quantum device. The much deeper quantum circuit of the standard QFIE version is indeed much more sensitive to the quantum noise of the *ibmq_sherbrooke* device than the optimized counterpart. The simplification of the optimized circuit (Fig. 7) with respect to the standard one (Fig. 6) actually produced more reliable results, with a control surface almost overlapping with the one from its ideal simulator in Fig. 9b, exhibiting a much smaller impact of noise on the results.

Table 2
RMSD and HD between the control surfaces obtained with the classical FRBS and the quantum alternatives.

	Standard QFIE		Optimized QFIE	
	RMSD	HD	RMSD	HD
Noiseless simulation	0.0079	0.2488	0.0078	0.2429
Noisy simulation	0.0724	0.3571	0.0179	0.2412
NISQ hardware	0.2569	1.6250	0.0271	0.2396

To explain the big difference between the control surface obtained with the standard QFIE approach in the noisy simulation scenario and in the execution on the real hardware, it is important to highlight that there can be important differences between the noisy model in the simulation and the noise that affects the computation on the real quantum hardware. As an example, considering the qubit labeled as 0 in the *ibmq_sherbrooke* processor, it is possible to see that the readout error on that qubit moves from a value of 0.0049[1] in the noise model of the simulation backend, to a value 0.0168[2] obtained from the calibration data of the actual quantum device.

To quantitatively assess the similarity between control surfaces, RMSD and HD metrics (Eqs. (36) and (37)) were computed, using the classical FRBS surface as a baseline. The results, summarized in Table 2, reveal that the optimized QFIE achieves an RMSD approximately 10 times lower than the standard QFIE when comparing the quantum hardware execution to the noiseless simulation, and 4 times lower when comparing the noisy simulations.

As expected, for the standard QFIE, these distance metrics increase accordingly with noise, being maximum for the surface computed using real quantum hardware. In contrast, the optimized QFIE maintains stable metrics across different execution environments, confirming its robustness against quantum noise and its suitability to be run on NISQ devices. As a final comparison between the two QFIE variants, we investigated how the number of shots affects the resulting control surfaces. Specifically, we evaluated the control surfaces produced by the Standard and Optimized QFIE using 500, 1000, 5000, and 10000 shots. Fig. 12 shows that, with respect to the chosen metrics – RMSD (a) and HD (b), computed using the classical surface as reference – the results are largely insensitive to the number of shots. This observation supports the choice of 1000 shots for both simulations and hardware executions.

5.2. Edge detection experiments with optimized QFIE-based FRBS

The second part of our experiments consists of evaluating the capability of the optimized QFIE-based FRBS in performing ED compared to the classical counterpart. In detail, the lookup tables shown as control surfaces in Section 5.1.2 are exploited to perform ED on the images from the BSDS-500 dataset. Firstly, Section 5.2.1 will introduce the metrics used to perform the aforementioned comparison. Section 5.2.2 will analyze the results obtained complemented by a comparison of the performances with Canny [13] and Sobel [14] methods. Finally, in Section 5.2.3, a study is presented on how the ED capability of a FRBS approach based on the lookup table changes according to the dimensionality of the table.

5.2.1. Metrics for image analysis

To assess the quality of the results of the proposed quantum FRBS approach in performing ED, three comparison metrics have been taken into account. These metrics are full-reference, i.e., an input image is compared against a reference image, which is assumed to be the ground truth. The first two metrics are *Sensitivity*

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \tag{38}$$

and *Specificity*

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}. \tag{39}$$

Both are binary metrics, hence for this purpose pixel intensity has been approximated to 0 (black) when below 0.5 and to 1 (white) vice versa. Positives and negatives are defined in Table 3.

Table 3
Binary classifiers definitions.

Ground truth	Prediction	Definition
0	0	True positive
0	1	False negative
1	0	False positive
1	1	True negative

In particular, Sensitivity (Eq. (38)) represents the ratio of how many edge pixels are correctly predicted over the total number of edge pixels in the ground truth image, evaluating the ability of the algorithm in exam to predict edge pixels on a scale from 0 (every edge pixel is wrongfully detected) to 1 (every edge pixel is correctly detected).

Specificity (Eq. (39)) represents the ratio of how many background pixels are correctly predicted over the total number of background pixels in the ground truth image, evaluating the ability of the algorithm in exam to predict background pixels on a scale from 0 (every background pixel is wrongfully detected) to 1 (every background pixel is correctly detected).

The third metric in the exam is the *structural similarity index measure (SSIM)*, a full reference metric introduced in [81]. It has been evaluated using the function *structural_similarity* of the python library scikit-image [82], and is defined as

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{40}$$

where:

- μ_x and μ_y are the mean intensity values of images x (ground-truth) and y,
- σ_x^2 and σ_y^2 are the variances of x and y,
- σ_{xy} is the covariance between x and y,
- C_1 and C_2 are small constants to stabilize the division.

This formula provides a measure of the similarity between two images by comparing luminance, contrast and structure, resulting in a more suited solution for ED problems with respect to other methods such as, for example, the well-known mean squared error [83]. In this sense, this metric approaches one in the limit where the two images compared are identical.

5.2.2. Results on the BSDS-500 dataset

This section aims to show the results of the experiments carried out by performing ED through the optimized QFIE-based FRBS on the images in the BSDS-500 dataset, which comprises 500 high-quality, heterogeneous natural images.

In detail, the ED was performed by inferring the edges from the control surfaces of the classical FRBS and the optimized QFIE-based FRBS when computed both using the ideal simulator (Fig. 9b), the noisy simulator (Fig. 11a) and the actual quantum machine (Fig. 11b). Some representative examples are presented in Figs. 13–16, where given the original image in the dataset (Figs. 13–16a), the edges were computed by means of the classical FRBS (Figs. 13–16b) and by noiseless (Figs. 13–16c) and noisy simulations (Figs. 13–16d) of QFIE. These

[1] This value is related to the date 27-05-2024.
[2] This value is related to the date 20-02-2025.

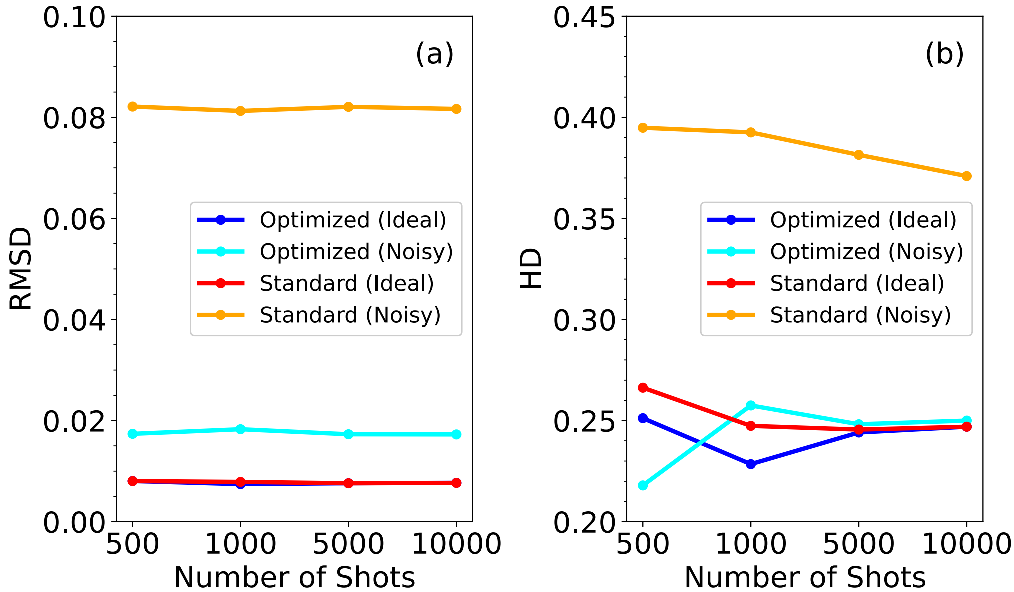


Fig. 12. RMSD (a) and HD (b) between the control surfaces obtained with the classical FRBS and the quantum alternatives for different numbers of shots.

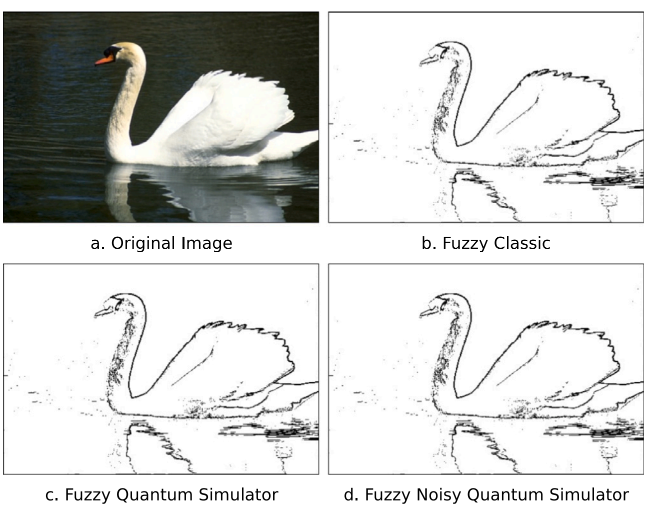


Fig. 13. ED on a photo of a swan (a) computed by means of the classical FRBS (b) by noiseless (c) and noisy simulations of the optimized QFIE (d).

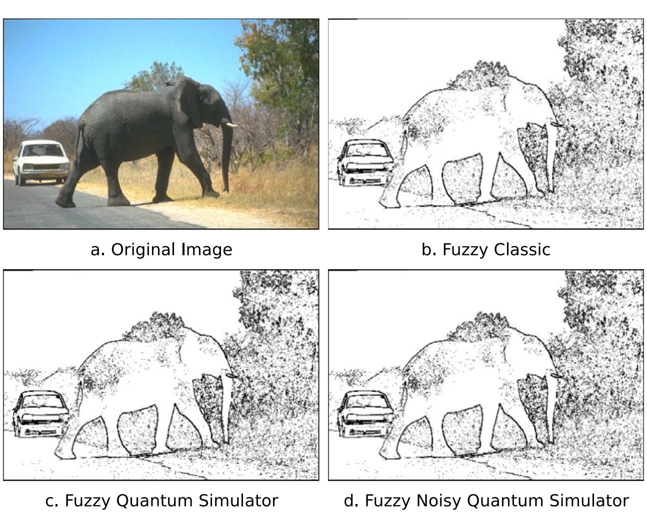


Fig. 14. Same as Fig. 13 on a photo of an elephant.

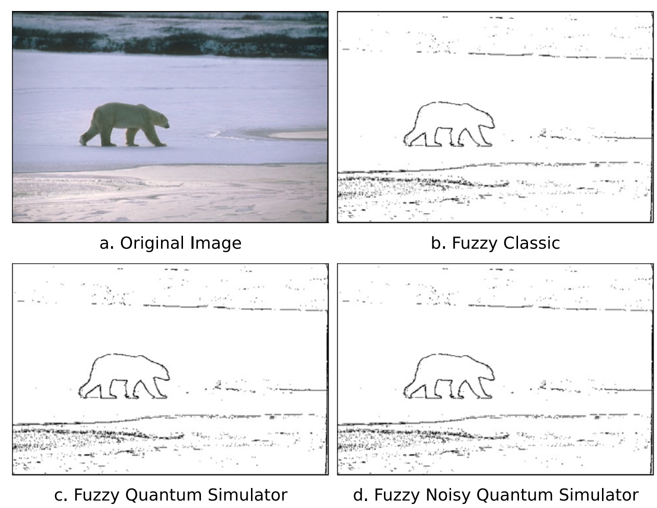
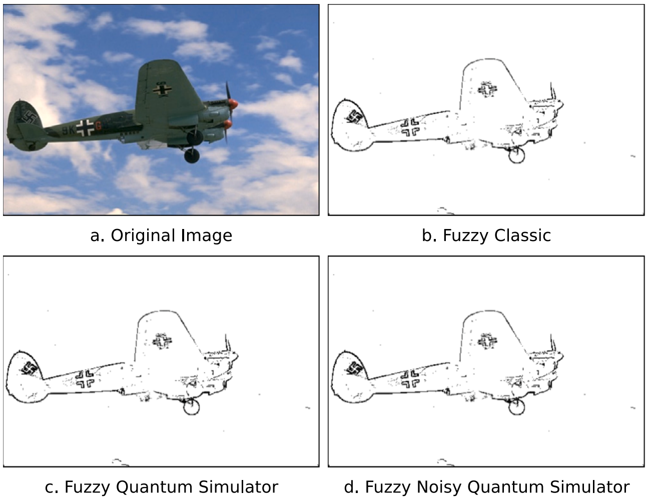


Fig. 15. Same as Fig. 13 on a photo of a polar bear.

results highlight the effectiveness of the proposed approach. In fact, the objects are precisely delineated with clear boundaries across diverse conditions, preserving considerable detail. From a visual point-of-view, the results obtained through quantum simulations appear virtually indistinguishable from classical implementations. The corresponding examples processed on the real quantum device are shown in Fig. 17. Remarkably, these results also appear visually consistent with both the classical and simulated quantum approaches.

For a quantitative comparison between the optimized QFIE implementations across different devices involved, using the classical approach as baseline (ground truth), we evaluated Sensitivity, Specificity, and SSIM metrics. Table 4 presents the mean, minimum, and maximum values computed across the entire dataset.

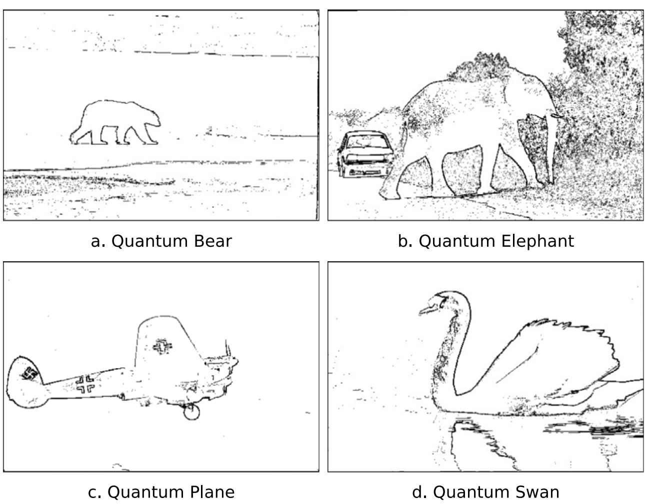
A striking observation is that Sensitivity consistently equals 1 across all configurations, indicating a complete absence of false negatives throughout the dataset — i.e. no edge pixel was misclassified as background. Equally noteworthy, Specificity also achieves perfect scores, implying the absence of false positives — i.e. no background pixels



a. Original Image b. Fuzzy Classic

c. Fuzzy Quantum Simulator d. Fuzzy Noisy Quantum Simulator

Fig. 16. Same as Fig. 13 on a photo of a plane.



a. Quantum Bear b. Quantum Elephant

c. Quantum Plane d. Quantum Swan

Fig. 17. ED for different cases on *ibmq-sherbrooke* NISQ hardware.

Table 4
Metrics on the whole dataset across different configurations and samples, compared with classical results for the optimized QFIE.

Metric	Configuration	Mean	Minimum	Maximum
Sensitivity	Simulator	1.0000	1.0000	1.0000
	Noisy	1.0000	1.0000	1.0000
	Quantum	1.0000	1.0000	1.0000
Specificity	Simulator	1.0000	1.0000	1.0000
	Noisy	1.0000	1.0000	1.0000
	Quantum	1.0000	1.0000	1.0000
SSIM	Simulator	0.9912	0.9793	0.9998
	Noisy	0.9910	0.9786	0.9997
	Quantum	0.9920	0.9809	0.9998

were misclassified as edges across the entire dataset. Furthermore, SSIM approaches 1 for all three configurations, highlighting strong compatibility between different platforms, even for the real NISQ machine.

The same experiment was carried out for the FRBS equipped with the standard QFIE. It also produced promising results; while Sensitivity remains constantly equal to 1, Specificity shows a mean value of 0.9339, ranging from 0.7156 to 0.9981. These results, while showcasing the presence of some false positives, still demonstrate remarkable performance. The SSIM metric yields similarly satisfactory results, ranging from 0.3705 to 0.9944, with a mean value of 0.7572 and minimal variability between methods.

To further evaluate the performance of the classical fuzzy engine and the QFIE, it is worth analyzing their behavior with respect to well-established ED methods. In particular, we focus on the Canny [13] and Sobel [14] algorithms.

We fixed the SSIM metric over the entire BSDS-500 dataset, using the ground truth annotations provided with the dataset to generate the box plot shown in Fig. 18. In this figure, we compare four different edge detection methods: Fuzzy Classic, Fuzzy Quantum (on *ibmq-sherbrooke*), Canny, and Sobel.

As shown, all four methods exhibit minimum SSIM values close to zero, corresponding to a subset of test images that are particularly challenging for gradient-based edge detection techniques, as the one reported in Fig. 19. The Fuzzy methods, however, reach near-maximum SSIM values, indicating that under optimal conditions, they are capable of producing very high-quality results. The absence of outliers in these methods further highlights their consistency across varying image conditions. The median SSIM values for both the Fuzzy Classical and Fuzzy Quantum approaches are nearly identical, reinforcing the reliability of the quantum model and its robustness to noise. Notably, their median values are also higher than those of the Canny and Sobel methods. Overall, both fuzzy-based techniques clearly outperform the traditional algorithms in terms of structural similarity. This result is further confirmed by the performance profiles illustrated in Fig. 20, which shows the distribution functions of the SSIM metric across the entire dataset, following the benchmarking approach developed in [84].

In more detail, these profiles have been obtained assuming as a performance measure of interest, for each image, the average value of the SSIM over all available ground truths, denoted as $\mathrm{SSIM}_{\mathrm{mean}}$. Since the SSIM values range in the interval $[0,1]$, with higher values indicating better similarity and image quality, we defined the performance metric as $(1-\mathrm{SSIM}_{\mathrm{mean}})$, in order to align with the minimization-based framework of [84]. For each image I and method s, we then computed the performance ratio $r_{I,s}$ as the ratio between $(1-\mathrm{SSIM}_{\mathrm{mean}})$ and the best (i.e. the minimum) value achieved across all methods, namely

$$r_{I,s} = \frac{(1-\mathrm{SSIM}_{\mathrm{mean}})_{I,s}}{\min\{(1-\mathrm{SSIM}_{\mathrm{mean}})_{I,s} : s \in S\}}, \tag{41}$$

where S denotes the set of solvers. The corresponding performance profile for each method s and threshold $\theta \geq 1$ is given by

$$\rho_s(\theta) = \frac{1}{n_I}\mathrm{size}\left\{I \in \mathcal{D} : r_{I,s} \leq \theta\right\} \tag{42}$$

where $\mathcal{D}$ and n_I denote, respectively, the dataset of images and its cardinality.

This function $\rho_s(\theta)$ represents the probability that method s achieves a performance within a factor θ of the best observed performance. In particular, $\rho(1)$ represents the probability that the corresponding solver will win over the others, in other words, it indicates the proportion of images for which method s delivers the best result. As shown in Fig. 20, both Fuzzy-based methods outperformed Sobel and Canny filters on over 90% of the images, with a very slight difference between Fuzzy classic and Fuzzy Quantum (less than 0.005).

5.2.3. *Analysis on the granularity of control surfaces*

As discussed in Section 4.4, an approach based on the construction of a lookup table has been exploited for using the QFIE-based approach on NISQ devices. This permits to build on time the lookup table, with a given granularity in the input space of the FRBS to perform then the inference on any image. The control surfaces showcased in Section 5.1.2 enabled the graphical visualization of the lookup tables obtained both for the classical FRBS as well as for the ones equipped with the standard and the optimized QFIE computed by using different backends. In detail, the said control surfaces have been computed receiving in input every possible couple of gradient values (I_x, I_y),

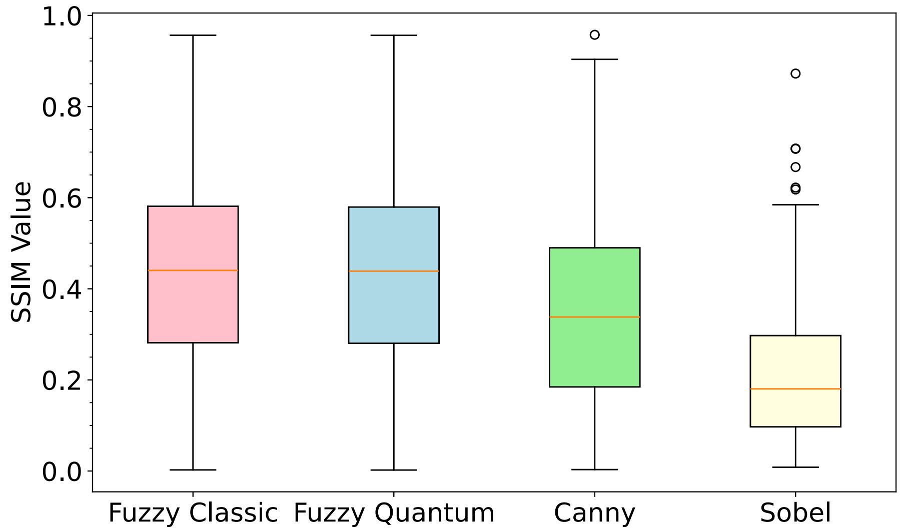


Fig. 18. Box plots for four different edge detection methods, Fuzzy Classic, Fuzzy Quantum, Canny and Sobel, using the SSIM metric on the whole BSDS-500 dataset adopting the ground truths provided by the dataset.

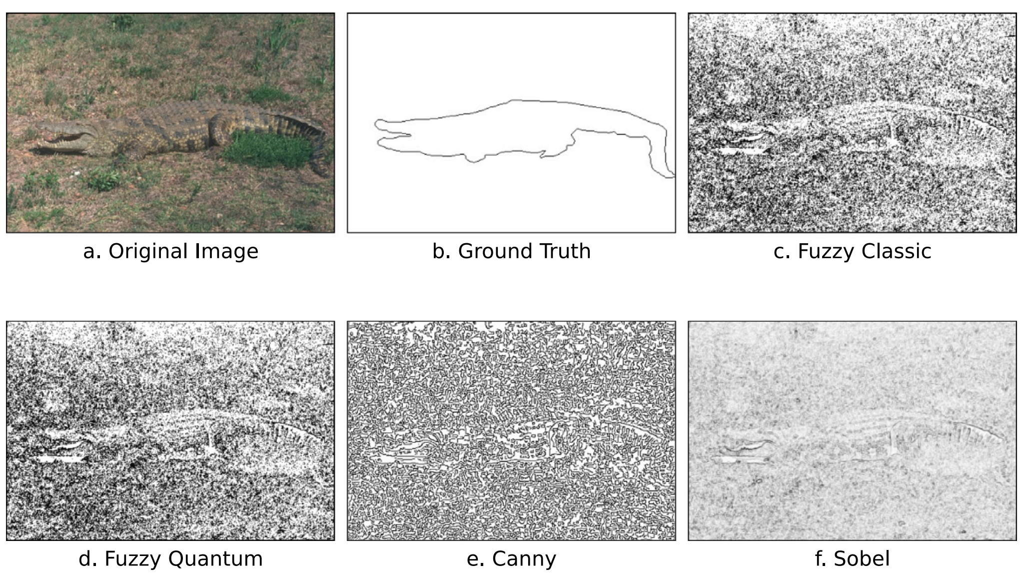


Fig. 19. Crocodile test image (a) with corresponding BSDS-500 ground truth (b) and corresponding edges determined with Fuzzy Classic (c), Fuzzy Quantum (d), Canny (e), and Sobel (f).

ranging from -1 to 1, with steps of 0.1. The goal of this section is to analyze how by changing the granularity of the input space in the construction of these lookup tables changes the capability of the FRBS in performing ED according to the metrics discussed in Section 5.2.1. To this end, it has been carried out a study by comparing the ED capability of a classical FRBS that performs ED on the whole BSDS-500 dataset performing the inference pixel by pixel for each image in the dataset, to a classical FRBS that performs the ED by inferring the output from a lookup table. In particular, it was varied the precision with which these lookup tables were built, by varying the granularity of the input space with four different values 0.1, 0.01, 0.001, 0.0001. Consequently, the dimensions of the lookup tables vary in these four scenarios from 441 inputs when the table was built by considering a step of 0.1 in the input space, to 441 000 inputs when considering a precision of 0.0001.

The average of Sensitivity (Eq. (38)), Specificity (Eq. (39)) and SSIM (Eq. (40)) metrics have been computed over the whole BSDS-500 dataset, assuming as truth values the images generated by the FRBS not based on the lookup table mechanism.

As shown in Fig. 21, the algorithm converges to the standard implementation for a precision on the second decimal place, while providing reasonable results also on the first decimal place. That said, the rationale behind the chosen precision on the first decimal place in the experiments discussed in Section 5.2.2 is two-fold: firstly, the metrics in exam corroborate reasonable accuracy results despite the lower precision, to the extent that the differences are negligible also from a visual point-of-view; additionally, the advantage of operating on a problem of a lower order of magnitude of complexity results crucial for the execution within a reasonable timing on the NISQ machine. In any case, this also implies that the reported performance constitutes a lower bound on what could be achieved with actual quantum execution. We plan to investigate the full quantum implementation in future work as hardware availability improves.

5.3. Noise-resilience analysis

To further validate the suitability of our model in being computed on NISQ devices, we now assess its noise resilience capability. Modern

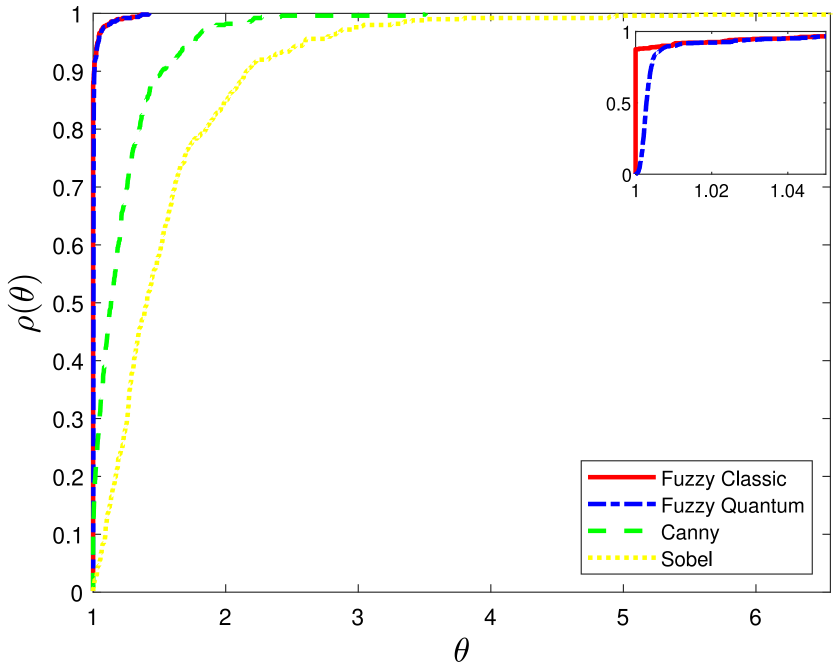


Fig. 20. Performance profiles of SSIM metric on the BSDS-500 dataset.

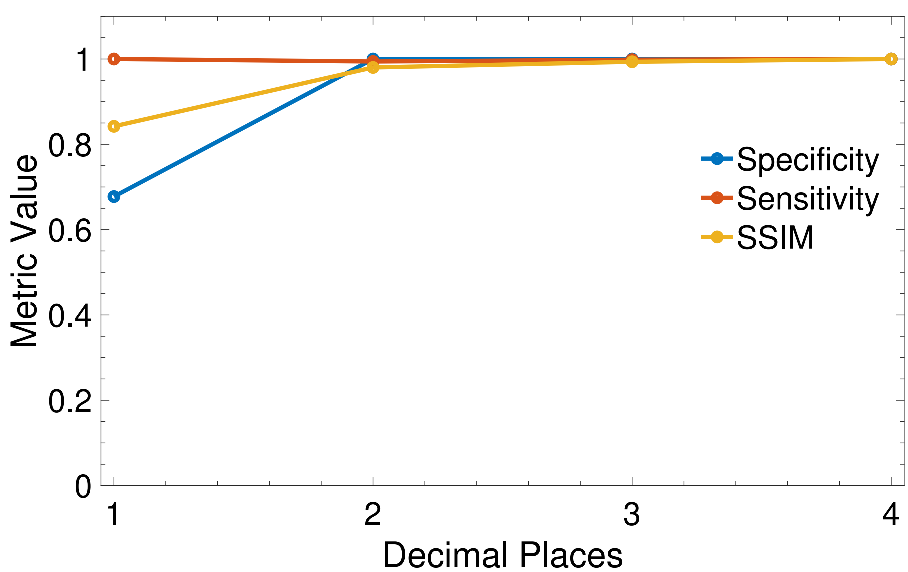


Fig. 21. Average metrics on the whole BSDS-500 dataset across different precisions, compared with classical results.

frameworks like Qiskit [85] and PennyLane [86] allow one to inject custom noise channels into quantum-circuit simulations. In this work, we restrict our noise model to two channels: the *readout* channel, which captures measurement errors, and the *depolarizing* channel, which represents imperfections in single- and two-qubit gates (Section 3.2). In particular, the latter has been applied to the basis gates composing our reference quantum device *ibmq-sherbrooke*, i.e. the single-qubit gates ID, RZ, SX, X and the two-qubits gate ECR.

To explore the impact of both error types, we vary the readout probability s_{ro} and the single-qubit depolarizing strength s_d over 25 uniformly spaced combinations in the intervals

$$s_{ro} \in [0, 0.25], \quad s_d \in [0, 0.02].$$

The two-qubit depolarizing strength, is fixed to the value of $10s_d$. These ranges reflect typical values reported for IBM quantum hardware (see for example [67]).

We generated a control surface for each pair of error probabilities (s_{ro}, s_d) and computed the RMSD of these surfaces relative to the ideal (noise-free) control surface, i.e. $s_{ro} = 0$ and $s_d = 0$. The outcome is shown in Fig. 22. As we can see, the RMSD increases with the increase of the noise up to a maximum value of ≈0.2. This is obtained in correspondence of the maximum values of (s_{ro}, s_d), namely $(s_{ro} = 0.25, s_d = 0.02)$. It is worth noticing that this maximum value is practically an upper limit for all NISQ machines currently available. In fact, on *ibmq sherbrooke*, the NISQ machine we have used, the readout channel, for example, ranges from 0.002 to 0.5 (only for 1 qubit), with a median value of 0.02, the Pauli-X error ranges from 7×10^{-5} to 9×10^{-3} with a median value of 2×10^{-4}, and, finally, the ECR error ranges from 3×10^{-3} to 1×10^{-1} with a median value of 6×10^{-3}. To better understand the impact of the error on the edge from a visual point-of-view, using the swan picture from the BSDS-500 dataset [12], in Fig. 23 we plot a comparison between the edges

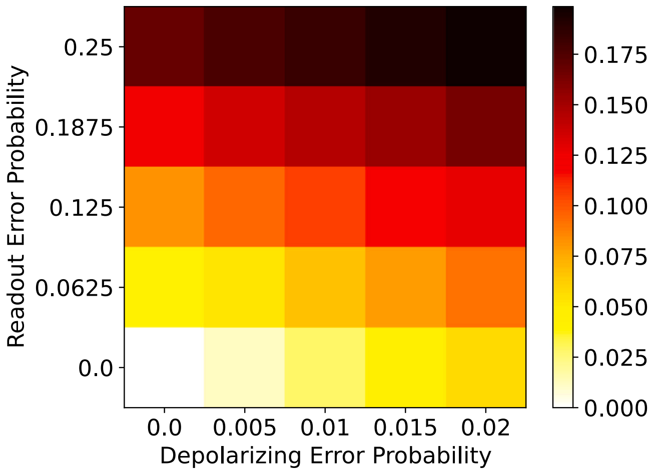


Fig. 22. RMSD values of the control surfaces for each pair of error probabilities (s_{ro}, s_d) with $s_{ro} \in [0, 0.25]$, $s_d \in [0, 0.02]$ computed relative to the ideal (noise-free) surface at $s_{ro} = 0$ and $s_d = 0$.

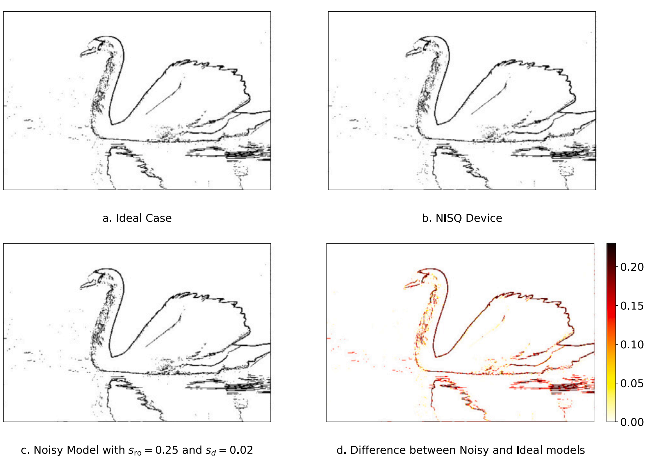


Fig. 23. Visual comparison between the edges obtained for the ideal case (a), with the quantum hardware *ibmq-sherbrooke* (b), with the noise model with the largest values of (s_{ro}, s_d) (c), and the absolute difference between the noisy and ideal models (d).

obtained for the ideal case (a), with the quantum hardware *ibmq-sherbrooke* (b), with the noise model with the largest values of (s_{ro}, s_d) (c), and the absolute difference between the noisy and ideal models (d). Remarkably, the results appear virtually indistinguishable despite the different noise levels, further substantiating the resilience of our model to quantum noise. To demonstrate this, we report the differences between the detected edges in Fig. 23(d), where the difference map is displayed using a standard linear colormap, where lighter red pixels indicate lower differences and darker red pixels highlight regions with greater deviation. As we can see, the differences between the detected edges are minimal, with a maximum pixel-wise intensity difference of approximately 0.23 on a normalized grayscale scale ranging from 0 to 1. These discrepancies primarily affect the gray intensity values of edge pixels and are hardly perceptible in visual inspection.

6. Conclusions

This study introduced a novel QFIE for edge detection implemented on NISQ hardware, demonstrating that quantum fuzzy systems can effectively process image gradients while respecting the constraints of NISQ devices. We designed an optimized QFIE circuit that significantly reduces the number of qubits and gate depth compared to the standard approach, making it more suitable for execution on current quantum processors. Remarkably, our results confirm that QFIE-based edge detection is not only feasible on quantum hardware but also more computationally efficient than its classical FRBS counterpart. While classical FRBS requires two rule evaluations per inference, the QFIE approach performs inference with a single quantum oracle query composed of just one multi-qubit quantum gate. To test the proposed quantum model on real-world images while dealing with the difficulty in accessing currently available quantum processors, a hybrid quantum–classical approach was adopted. It uses a pre-computed lookup table for mitigating the execution overhead of quantum circuit evaluations. This strategy enabled practical implementation on NISQ devices without compromising performance. These limitations will be overcome as quantum devices improve in terms of broader accessibility to quantum resources. Specifically, the lookup table approach can be replaced by direct pixel-wise quantum inference, making large-scale applications more practically viable. In this context, we also plan to investigate the possibility of processing multiple pixels in parallel, leveraging large QPUs. In principle, it would be possible to process up to $int(\mathrm{N}_{\mathrm{qubit}}/3)$ pixels simultaneously on a quantum processor with $\mathrm{N}_{\mathrm{qubit}}$ qubits. However, using a considerable number of qubits introduces additional errors, such as cross-talk between qubits, which can lead to unwanted entanglement and reduced fidelity of the computation. We therefore aim to carefully analyze the trade-off between the degree of parallelization and the cumulative noise, in order to optimize performance while mitigating error propagation.

As proven by the experiments carried out on the BSDS-500 dataset, the optimized QFIE circuit combined with this lookup table approach achieves the aforementioned advantage while maintaining accuracy comparable to classical methods, even under the limitations of real quantum hardware.

It is worth pointing out that, although we used a simple fuzzy set for the inputs and output, this simplification allowed us to streamline the structure of the QFIE, reducing the number of required gates and operations — a crucial advantage when working with limited quantum resources. In fact, this streamlined fuzzy configuration enabled the design of an optimized quantum circuit that significantly reduces the quantum resource requirements, especially after transpilation. Before transpilation, the optimized QFIE circuit already featured a simpler structure compared to the standard one. After transpilation, when targeting the *ibmq_sherbrooke* quantum processor, the circuit achieves a 78% reduction in depth (from 198 to 43) and an 80% reduction in the number of two-qubit gates (from 44 to 9).

Despite these simple fuzzy sets, the quality of our results significantly outperforms well-established classical methods such as Canny and Sobel edge detectors. While it is plausible that introducing a finer fuzzy set granularity – e.g., using three or more membership functions – could lead to improved performance, this hypothesis remains to be demonstrated. We plan to investigate this direction in future work.

In any case, we emphasize that the standard QFIE design [11] is flexible and can accommodate a larger number of fuzzy sets if required by more complex applications or future implementations.

These results represent a significant step toward exploiting quantum computing for soft computing applications, particularly in image processing tasks constrained by classical hardware limitations. Future work will focus on extending this approach to more complex image-processing tasks as well as on investigating the performance of a more expressive fuzzy system.

CRediT authorship contribution statement

G. Nunziata: Writing – review & editing, Writing – original draft, Software, Investigation. **S. Crisci:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **G. De Gregorio:** Writing – review & editing, Writing – original draft, Supervision, Data curation, Conceptualization. **R. Schiattarella:** Writing

– review & editing, Writing – original draft, Software, Methodology. **G. Acampora:** Visualization, Validation, Supervision. **L. Coraggio:** Visualization, Validation, Funding acquisition. **N. Itaco:** Writing – review & editing, Visualization, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

G. De Gregorio and S. Crisci acknowledge the support by EU-FESR, PON Ricerca e Innovazione 2014-2020- DM 1062/2021, and the collaboration with Giovanni Frattini, and the company partner of the project Engineering Ingegneria Informatica. S. Crisci aknolwedges the partial support of the project FEEDING of the program "Giovani Ricercatori" of University of Campania "L. Vanvitelli" and the National Group for Scientific Computing (INdAM-GNCS). L. Coraggio and N. Itaco acknowledge the financial support of the European Union - Next Generation EU, Mission 4 Component 1, CUP I53D23001060006, for the PRIN 2022 project "Exploiting separation of scales in nuclear structure and dynamics". G. Nunziata acknowledges the MIUR-PRIN 2022 Project "Numerical Optimization with Adaptive Accuracy and Applications to Machine Learning", project code: 2022N3ZNAX (CUP E53D2300 7700006), under the National Recovery and Resilience Plan (PNRR), Italy, Mission 04 Component 2 Investment 1.1 funded by the European Commission. - NextGeneration EU programme Research. R. Schiattarella was supported by the project "National Centre for HPC, Big Data and Quantum Computing" (PNRR CN00000013, CUP E63C22000980007), funded under the PNRR, Mission 4, Component 2, Investment 1.4. G. Acampora was supported by the PNRR MUR project PE0000013-FAIR. This work was funded by the European Union - NextGenerationEU under the project PNRR "National Centre for HPC, Big Data and Quantum Computing (HPC)" CN00000013 (CUP D43C22001240001) [MUR Decree n. 1031- 17/06/2022] - Cascade Call launched by SPOKE 10 POLIMI: "QML-NTED" project.

Data availability

The code for replicating our experiments is available at: https://github.com/gnunziata42/QFED.git.

References

[1] Z. Wang, M. Xu, Y. Zhang, Review of quantum image processing, Arch. Comput. Methods Eng. 29 (2) (2022) 737–761.
[2] M.C. Shin, D.B. Goldgof, K.W. Bowyer, Comparison of edge detector performance through use in an object recognition task, Comput. Vis. Image Underst. 84 (1) (2001) 160–178.
[3] R. Muthukrishnan, M. Radha, Edge detection techniques for image segmentation, Int. J. Comput. Sci. Inf. Technol. 3 (6) (2011) 259.
[4] S.J.H. Pirzada, A. Siddiqui, Analysis of edge detection algorithms for feature extraction in satellite images, in: 2013 IEEE International Conference on Space Science and Communication, IconSpace, IEEE, 2013, pp. 238–242.
[5] F. Yan, S.E. Venegas-Andraca, K. Hirota, Toward implementing efficient image processing algorithms on quantum computers, Soft Comput. 27 (18) (2023) 13115–13127.
[6] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum 2 (2018) 79.
[7] M.I. Chacón M, Fuzzy logic for image processing: Definition and applications of a fuzzy image processing scheme, in: Advanced Fuzzy Logic Technologies in Industrial Applications, Springer, 2006, pp. 101–113.
[8] C.-W. Tao, W.E. Thompson, J. Taur, A fuzzy if-then approach to edge detection, in: [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, IEEE, 1993, pp. 1356–1360.
[9] Z. Talai, A. Talai, A fast edge detection using fuzzy rules, in: 2011 International Conference on Communications, Computing and Control Applications, CCCA, IEEE, 2011, pp. 1–5.
[10] L. Hu, H.-D. Cheng, M. Zhang, A high performance edge detector based on fuzzy inference rules, Inform. Sci. 177 (21) (2007) 4768–4784.
[11] G. Acampora, R. Schiattarella, A. Vitiello, On the implementation of fuzzy inference engines on quantum computers, IEEE Trans. Fuzzy Syst. 31 (5) (2023) 1419–1433, http://dx.doi.org/10.1109/TFUZZ.2022.3202348.
[12] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916.
[13] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698, http://dx.doi.org/10.1109/TPAMI.1986.4767851.
[14] I. Sobel, G. Feldman, A 3x3 isotropic gradient operator for image processing, 1968, Presented at the Stanford Artificial Intelligence Project (SAIL).
[15] R. Sun, T. Lei, Q. Chen, Z. Wang, X. Du, W. Zhao, A.K. Nandi, Survey of image edge detection, Front. Signal Process. 2 (2022) http://dx.doi.org/10.3389/frsip.2022.826967.
[16] J.M.S. Prewitt, Object enhancement and extraction, in: B. Lipkin, A. Rosenfeld (Eds.), Picture Processing and Psychopictorics, Academic Press, New York, 1970, pp. 75–149.
[17] L.G. Roberts, Machine Perception of Three-Dimensional Solids (Ph.D. thesis), Massachusetts Institute of Technology, Cambridge, MA, 1963.
[18] D. Marr, E. Hildreth, Theory of edge detection, Proc. R. Soc. 207 (1980) 187–217.
[19] P. Marthon, B. Thiesse, A. Bruel, Edge detection by differences of Gaussians, in: Computer Vision for Robots, SPIE, 1986, pp. 318–327.
[20] S.M. Bhandarkar, Y. Zhang, W.D. Potter, An edge detection technique using genetic algorithm-based optimization, Pattern Recognit. 27 (9) (1994) 1159–1180, http://dx.doi.org/10.1016/0031-3203(94)90003-5.
[21] Y. Yongbin, Y. Chenyu, D. Quanxin, N. Tashi, L. Shouyi, Z. Chen, Memristive network-based genetic algorithm and its application to image edge detection, J. Syst. Eng. Electron. 32 (5) (2021) 1062–1070, http://dx.doi.org/10.23919/JSEE.2021.000091.
[22] Y.-H. Kuo, C.-S. Lee, C.-C. Liu, A new fuzzy edge detection method for image enhancement, in: Proceedings of 6th International Fuzzy Systems Conference, Vol. 2, IEEE, 1997, pp. 1069–1074.
[23] F. Russo, Edge detection in noisy images using fuzzy reasoning, IEEE Trans. Instrum. Meas. 47 (5) (1998) 1102–1105.
[24] L. Hua, H.D. Cheng, M. Zhang, A high performance edge detector based on fuzzy inference rules, J. Inf. Sci. 177 (21) (2007) 4768–4784.
[25] O. Mendoza, P. Melin, G. Licea, Fuzzy inference systems type-1 and type-2 for digital images edges detection, Eng. Lett. 15 (1) (2007).
[26] O. Mendoza, P. Melin, G. Licea, A new method for edge detection in image processing using interval type-2 fuzzy logic, in: 2007 IEEE International Conference on Granular Computing, GRC 2007, IEEE, 2007, p. 151.
[27] O. Mendoza, P. Melin, The fuzzy Sugeno integral as a decision operator in the recognition of images with modular neural networks, in: Hybrid Intelligent Systems Design and Analysis, Springer-Verlag, Germany, 2007, pp. 299–310.
[28] O. Mendoza, P. Melin, O. Castillo, G. Licea, Type-2 fuzzy logic for improving training data and response integration, in: Modular Neural Networks for Image Recognition, Foundations of Fuzzy Logic and Soft Computing, LNCC, Springer-Verlag, Germany, 2007, pp. 604–612.
[29] O. Mendoza, P. Melin, G. Licea, A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks, and the Sugeno integral, J. Inf. Sci. 179 (13) (2009) 2078–2101.
[30] P. Melin, O. Mendoza, O. Castillo, An improved method for edge detection based on interval type-2 fuzzy logic, Expert Syst. Appl. 37 (12) (2010) 8527–8535.
[31] J. Yang, B. Price, S. Cohen, Object contour detection with a fully convolutional encoder-decoder network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 193–202, http://dx.doi.org/10.1109/CVPR.2016.28.
[32] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, CoRR abs/1412.6980.
[33] Z. Yu, C. Feng, M.-Y. Liu, S. Ramalingam, Casenet: Deep category-aware semantic edge detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5964–5973.
[34] M. Pu, Y. Huang, Q. Guan, RINDNet: Edge detection for discontinuity in reflectance, illumination, normal, and depth, 2021, Arxiv Preprint arxiv:2106.00616.
[35] P.Q. Le, F. Dong, K. Hirota, A flexible representation of quantum images for polynomial preparation, image compression, and processing operations, Quantum Inf. Process. 10 (1) (2011) 63–84.
[36] Y. Zhang, K. Lu, Y. Gao, M. Wang, NEQR: a novel enhanced quantum representation of digital images, Quantum Inf. Process. 12 (2013) 27, http://dx.doi.org/10.1007/s11128-013-0567-z.
[37] L. Tang, Image edge detection based on quantum genetic algorithm, Comput. Model. New Technol. 18 (12B) (2014) 517–523.
[38] X.-W. Yao, H. Wang, Z. Liao, M.-C. Chen, J. Pan, J. Li, K. Zhang, X. Lin, Z. Wang, Z. Luo, W. Zheng, J. Li, M. Zhao, X. Peng, D. Suter, Quantum image processing and its application to edge detection: Theory and experiment, Phys. Rev. X 7 (2017) 031041, http://dx.doi.org/10.1103/PhysRevX.7.031041.

[39] R.-G. Zhou, D.-Q. Liu, Quantum image edge extraction based on improved Sobel operator, Internat. J. Theoret. Phys. 58 (9) (2019) 2969–2985, http://dx.doi.org/10.1007/s10773-019-04127-5.

[40] P. Fan, R.-G. Zhou, W.-W. Hu, N.-H. Jiang, Quantum image edge extraction based on Laplacian operator and zero-cross method, Quantum Inf. Process. 18 (2019) 27, http://dx.doi.org/10.1007/s11128-018-2129-x.

[41] P. Li, T. Shi, A. Lu, et al., Quantum implementation of classical Marr–Hildreth edge detection, Quantum Inf. Process. 19 (2020) 64, http://dx.doi.org/10.1007/s11128-019-2559-0.

[42] Y. Ma, H. Ma, P. Chu, Demonstration of quantum image edge extraction enhancement through improved Sobel operator, IEEE Access 8 (2020) 210277–210285, http://dx.doi.org/10.1109/ACCESS.2020.3038891.

[43] Z. Gao, S. Yang, et al., Quantum image edge detection using improved Sobel mask based on NEQR, Quantum Inf. Process. 20 (1) (2021) 1–18, http://dx.doi.org/10.1007/s11128-020-02944-7.

[44] C. Wu, F. Huang, J. Dai, N. Zhou, Quantum SUSAN edge detection based on double chains quantum genetic algorithm, Phys. A 605 (2022) 128017, http://dx.doi.org/10.1016/j.physa.2022.128017.

[45] A. Geng, A. Moghiseh, C. Redenbach, K. Schladitz, A hybrid quantum image edge detector for the NISQ era, Quantum Mach. Intell. 4 (2022) 15, http://dx.doi.org/10.1007/s42484-022-00071-3.

[46] M. Lisnichenko, S. Protasov, Quantum image representation: a review, Quantum Mach. Intell. 5 (1) (2022) 2, http://dx.doi.org/10.1007/s42484-022-00089-7.

[47] S. Chakraborty, S.B. Mandal, S.H. Shaikh, Quantum image processing: challenges and future research issues, Int. J. Inf. Technol. 14 (1) (2022) 475–489, http://dx.doi.org/10.1007/s41870-018-0227-8.

[48] R. Ur Rasool, H.F. Ahmad, W. Rafique, A. Qayyum, J. Qadir, Z. Anwar, Quantum computing for healthcare: A review, Futur. Internet 15 (3) (2023).

[49] H. Defienne, W.P. Bowen, M. Chekhova, G.B. Lemos, D. Oron, S. Ramelow, N. Treps, D. Faccio, Advances in quantum imaging, Nat. Photonics 18 (10) (2024) 1024–1036, http://dx.doi.org/10.1038/s41566-024-01516-w.

[50] M. Schuld, M. Fingerhuth, F. Petruccione, Implementing a distance-based classifier with a quantum interference circuit, Europhys. Lett. 119 (6) (2017) 60002, http://dx.doi.org/10.1209/0295-5075/119/60002.

[51] M. Henderson, S. Shakya, S. Pradhan, T. Cook, Quanvolutional neural networks: powering image recognition with quantum circuits, Quantum Mach. Intell. 2 (1) (2020) 2.

[52] J. Liu, K.H. Lim, K.L. Wood, W. Huang, C. Guo, H.-L. Huang, Hybrid quantum-classical convolutional neural networks, Sci. China Phys. Mech. Astron. 64 (9) (2021) 290311.

[53] D. Bokhan, A.S. Mastiukova, A.S. Boev, D.N. Trubnikov, A.K. Fedorov, Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning, Front. Phys. 10 (2022) http://dx.doi.org/10.3389/fphy.2022.1069985, Original Research Article, Section: Quantum Engineering and Technology.

[54] E.H. Houssein, Z. Abohashima, M. Elhoseny, W.M. Mohamed, Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images, J. Comput. Des. Eng. 9 (2) (2022) 343–363.

[55] W. Li, P.-C. Chu, G.-Z. Liu, Y.-B. Tian, T.-H. Qiu, S.-M. Wang, An image classification algorithm based on hybrid quantum classical convolutional neural network, Quantum Eng. 2022 (1) (2022) 5701479.

[56] Y. Dong, Y. Fu, H. Liu, X. Che, L. Sun, Y. Luo, An improved hybrid quantum-classical convolutional neural network for multi-class brain tumor MRI classification, J. Appl. Phys. 133 (6) (2023) 064401.

[57] S.-Y. Huang, W.-J. An, D.-S. Zhang, N.-R. Zhou, Image classification and adversarial robustness analysis based on hybrid quantum–classical convolutional neural network, Opt. Commun. 533 (2023) 129287, http://dx.doi.org/10.1016/j.optcom.2023.129287, URL https://www.sciencedirect.com/science/article/pii/S0030401823000329.

[58] L.-H. Gong, J.-J. Pei, T.-F. Zhang, N.-R. Zhou, Quantum convolutional neural network based on variational quantum circuits, Opt. Commun. 550 (2024) 129993, http://dx.doi.org/10.1016/j.optcom.2023.129993, URL https://www.sciencedirect.com/science/article/pii/S0030401823007411.

[59] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, J.-W. Pan, Experimental quantum generative adversarial networks for image generation, Phys. Rev. Appl. 16 (2021) 024051, http://dx.doi.org/10.1103/PhysRevApplied.16.024051, URL https://link.aps.org/doi/10.1103/PhysRevApplied.16.024051.

[60] N.-R. Zhou, T.-F. Zhang, X.-W. Xie, J.-Y. Wu, Hybrid quantum–classical generative adversarial networks for image generation via learning discrete distribution, Signal Process., Image Commun. 110 (2023) 116891, http://dx.doi.org/10.1016/j.image.2022.116891, URL https://www.sciencedirect.com/science/article/pii/S0923596522001709.

[61] L. Antonelli, V. De Simone, D. di Serafino, A view of computational models for image segmentation, Ann. Univ. Ferrara 68 (2) (2022) 277–294.

[62] A.J. Pinho, L.B. Almeida, A review on edge detection based on filtering and differentiation, Eletrónica Telecomunicações 2 (1) (1997).

[63] D. Mumford, J. Shah, Boundary detection by minimizing functionals, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 17, San Francisco, 1985, pp. 137–154.

[64] J.F. Canny, A variational approach to edge detection, in: AAAI, Vol. 1983, 1983, pp. 54–58.

[65] P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in: Proceedings 35th Annual Symposium on Foundations of Computer Science, Ieee, 1994, pp. 124–134.

[66] L.K. Grover, A framework for fast quantum mechanical algorithms, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 1998, pp. 53–62.

[67] G. Acampora, R. Schiattarella, On the effect of quantum noise in quantum genetic algorithms, in: 2023 IEEE Congress on Evolutionary Computation, CEC, 2023, pp. 1–8, http://dx.doi.org/10.1109/CEC53210.2023.10254078.

[68] Z. Kovacic, S. Bogdan, Fuzzy Controller Design: Theory and Applications, CRC Press, 2018.

[69] T.M. Inc., Fuzzy logic image processing, 2024, URL https://www.mathworks.com/help/fuzzy/fuzzy-logic-image-processing.html.

[70] V.V. Shende, S.S. Bullock, I.L. Markov, Synthesis of quantum logic circuits, in: Proceedings of the 2005 Asia and South Pacific Design Automation Conference, 2005, pp. 272–275.

[71] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, 2023, arXiv:2304.02643.

[72] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, J. Wang, Fast segment anything, 2023, arXiv:2306.12156.

[73] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1395–1403.

[74] X. Soria, G. Pomboza-Junez, A.D. Sappa, LDC: Lightweight dense CNN for edge detection, IEEE Access 10 (2022) 68281–68290, http://dx.doi.org/10.1109/ACCESS.2022.3186344.

[75] Z. Su, J. Zhang, L. Wang, H. Zhang, Z. Liu, M. Pietikäinen, L. Liu, Lightweight pixel difference networks for efficient visual representation learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (12) (2023) 14956–14974, http://dx.doi.org/10.1109/TPAMI.2023.3300513.

[76] Z. Luo, F.K. Gustafsson, Z. Zhao, J. Sjölund, T.B. Schön, Image restoration with mean-reverting stochastic differential equations, Int. Conf. Mach. Learn. (2023).

[77] J. Jiang, K. Zhang, R. Timofte, Towards flexible blind JPEG artifacts removal, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4997–5006.

[78] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: European Conference on Computer Vision, ECCV, Springer, 2014, pp. 740–755.

[79] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal Visual Object Classes (VOC) challenge, Int. J. Comput. Vis. (IJCV) 88 (2) (2010) 303–338.

[80] G. Acampora, A. Massa, R. Schiattarella, A. Vitiello, Distributing fuzzy inference engines on quantum computers, in: 2023 IEEE International Conference on Fuzzy Systems, FUZZ, 2023, pp. 1–6, http://dx.doi.org/10.1109/FUZZ52849.2023.10309786.

[81] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[82] S. Van der Walt, J.L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. Warner, N. Yager, E. Gouillart, V. Stéfan, Scikit-image: Image processing in Python, 2025, https://scikit-image.org, (Accessed 10 March 2025).

[83] Z. Wang, A.C. Bovik, Mean squared error: Love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 26 (1) (2009) 98–117, http://dx.doi.org/10.1109/MSP.2008.930649.

[84] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.

[85] A. Javadi-Abhari, M. Treinish, K. Krsulich, C.J. Wood, J. Lishman, J. Gacon, S. Martiel, P.D. Nation, L.S. Bishop, A.W. Cross, B.R. Johnson, J.M. Gambetta, Quantum computing with Qiskit, 2024, http://dx.doi.org/10.48550/arXiv.2405.08810, arXiv:2405.08810.

[86] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, et al., PennyLane: Automatic differentiation of hybrid quantum-classical computations, 2022, arXiv:1811.04968. URL https://arxiv.org/abs/1811.04968.