computation

IMPACT FACTOR 1.9

CITESCORE 4.1

Article

A Hybrid Quantum–Classical Architecture with Data Re-Uploading and Genetic Algorithm Optimization for Enhanced Image Classification

Aksultan Mukhanbet and Beimbet Daribayev

MDPI

https://doi.org/10.3390/computation13080185 *computation* MDPI

Article

A Hybrid Quantum–Classical Architecture with Data Re-Uploading and Genetic Algorithm Optimization for Enhanced Image Classification

Aksultan Mukhanbet [1,2,*] and Beimbet Daribayev [1,3]

[1] LLP "DigitAlem", Almaty 050042, Kazakhstan; beimbet.daribayev@gmail.com
[2] Faculty of Information Technology, Al-Farabi Kazakh National University, Almaty 050040, Kazakhstan
[3] Graduate School of Digital Technologies and Construction, Shakarim University, Semey 071412, Kazakhstan
[*] Correspondence: mukhanbetaksultan0414@gmail.com

Abstract

Quantum machine learning (QML) has emerged as a promising approach for enhancing image classification by exploiting quantum computational principles such as superposition and entanglement. However, practical applications on complex datasets like CIFAR-100 remain limited due to the low expressivity of shallow circuits and challenges in circuit optimization. In this study, we propose HQCNN–REGA—a novel hybrid quantum–classical convolutional neural network architecture that integrates data re-uploading and genetic algorithm optimization for improved performance. The data re-uploading mechanism allows classical inputs to be encoded multiple times into quantum states, enhancing the model's capacity to learn complex visual features. In parallel, a genetic algorithm is employed to evolve the quantum circuit architecture by optimizing gate sequences, entanglement patterns, and layer configurations. This combination enables automatic discovery of efficient parameterized quantum circuits without manual tuning. Experiments on the MNIST and CIFAR-100 datasets demonstrate state-of-the-art performance for quantum models, with HQCNN–REGA outperforming existing quantum neural networks and approaching the accuracy of advanced classical architectures. In particular, we compare our model with classical convolutional baselines such as ResNet-18 to validate its effectiveness in real-world image classification tasks. Our results demonstrate the feasibility of scalable, high-performing quantum–classical systems and offer a viable path toward practical deployment of QML in computer vision applications, especially on noisy intermediate-scale quantum (NISQ) hardware.

Keywords: quantum machine learning; hybrid quantum–classical models; data re-uploading; genetic algorithm; quantum convolutional neural network; image classification; circuit optimization

check for updates

Academic Editor: Xiaoqiang Hua

Received: 30 June 2025
Revised: 25 July 2025
Accepted: 26 July 2025
Published: 1 August 2025

Citation: Mukhanbet, A.; Daribayev, B.. A Hybrid Quantum–Classical Architecture with Data Re-Uploading and Genetic Algorithm Optimization for Enhanced Image Classification. *Computation* **2025**, *13*, 185. https://doi.org/10.3390/computation13080185

1. Introduction

The rapid advancement of quantum computing has sparked significant interest in quantum machine learning (QML), a field that promises to harness quantum phenomena—such as superposition, entanglement, and interference—to enhance classical computational paradigms [1]. Among its applications, image classification stands out as a benchmark task, with datasets like MNIST, CIFAR-10, and CIFAR-100 serving as standard tests for evaluating algorithmic performance [2,3]. While classical convolutional neural networks (CNNs) have achieved remarkable success, often exceeding 95% accuracy on CIFAR-100 classification tasks [4], their computational complexity and resource demands motivate the exploration of quantum alternatives that could potentially offer exponential speedups and improved efficiency [5].

Quantum convolutional neural networks (QCNNs), leveraging parameterized quantum circuits (PQCs), offer a potential pathway to efficiency and scalability, particularly as NISQ devices become more accessible [6,7]. The theoretical foundations of quantum machine learning suggest that quantum algorithms could provide computational advantages for certain machine learning tasks, including pattern recognition and feature extraction [8,9]. However, despite these theoretical promises, practical implementations of QCNNs have struggled to match classical benchmarks, especially on complex datasets like CIFAR-100, where accuracies typically range between 60 and 80% [10,11].

The field of quantum machine learning has evolved significantly since its inception, beginning with the early work of Biamonte et al. [12] and Dunco and Briegel [13], which laid the theoretical foundation for quantum learning algorithms. These pioneering studies demonstrated that quantum computers and quantum simulators [14] could potentially provide advantages in solving differential equations and machine learning problems due to quantum parallelism and the ability to process information in superposition states. Schuld et al. [15] took this concept further by introducing quantum feature maps, which allow classical data to be encoded into quantum states for processing by quantum algorithms.

The development of variational quantum algorithms has been particularly influential in bridging classical optimization with quantum computation. Farhi et al. [16] introduced the concept of variational quantum eigensolvers (VQEs), which laid the groundwork for parameterized quantum circuits in machine learning applications. This approach was subsequently extended by Benedetti et al. [17], who demonstrated that parameterized quantum circuits could serve as universal function approximators, similar to classical neural networks but potentially with greater expressivity in certain problem domains.

The adaptation of classical convolutional neural network architectures to the quantum domain has been a subject of intensive research. Cong et al. [18] pioneered the concept of quantum convolutional neural networks, introducing quantum convolution and pooling operations that maintain translational invariance while leveraging quantum superposition. Their work demonstrated the feasibility of quantum convolution on small-scale problems, achieving competitive performance on simplified versions of classical datasets. Henderson et al. [19] advanced this with multi-scale quantum convolutions capturing long-range correlations through entanglement. Liu et al. [20] introduced hybrid quantum–classical approaches combining quantum feature extraction with classical post-processing, establishing the dominant paradigm balancing quantum advantages with classical optimization effectiveness.

A significant breakthrough in quantum machine learning came with the introduction of data re-uploading techniques by Pérez-Salinas et al. [21]. This approach addresses the fundamental limitation of single-pass data encoding in quantum circuits by allowing multiple injections of classical data throughout the circuit. The re-uploading strategy dramatically increases the expressivity of parameterized quantum circuits, enabling them to approximate complex functions that would be impossible to represent with single-layer quantum circuits.

García-Martín et al. [22] extended the re-uploading concept by analyzing its theoretical properties and demonstrating its universal approximation capabilities. Their work provided rigorous theoretical foundations for understanding why re-uploading enhances quantum circuit expressivity and established bounds on the approximation error for various function classes. Schuld and Sweke [23] explored data encoding strategies in quantum machine learning, highlighting that re-uploading improves generalization but may cause barren plateaus if not properly designed.

The use of evolutionary algorithms in quantum circuit design has shown promise for automating quantum algorithm discovery. Spector et al. [24] pioneered the application of genetic programming to evolve quantum circuits, laying the groundwork for this approach. Malossini et al. [25] advanced this by tailoring genetic algorithms to optimize both structure and parameters, achieving circuits that outperformed manual designs. Cincio et al. [26] later applied machine learning and evolutionary strategies to variational quantum algorithms, demonstrating their ability to navigate complex landscapes and avoid local minima better than gradient-based methods.

Despite the theoretical promise of quantum approaches to image classification, several practical challenges have limited their real-world performance. Schuld et al. [23] identified the "curse of dimensionality" in quantum machine learning, where the exponential scaling of quantum state spaces can lead to sampling complexity issues. This challenge is particularly acute in image classification, where high-dimensional input data must be efficiently encoded into quantum states. The barren plateau phenomenon, extensively studied by McClean et al. [27], represents another significant challenge for quantum machine learning. In this phenomenon, the gradients of parameterized quantum circuits become exponentially small as the circuit depth increases, making training extremely difficult. This issue has been particularly problematic for deep quantum circuits designed for image classification tasks.

Noise in current NISQ devices presents additional challenges for quantum image classification. Preskill [6] outlined the limitations of NISQ devices and their impact on quantum algorithm performance. The presence of decoherence and gate errors can significantly degrade the performance of quantum machine learning algorithms, particularly those requiring deep circuits or long coherence times.

The limitations of current quantum hardware have shifted attention toward hybrid quantum–classical approaches. Mitarai et al. [28] analyzed various hybrid architectures, showing that integrating quantum and classical components can outperform purely quantum systems on today's hardware. Cerezo et al. [29] reviewed variational quantum algorithms, noting their promise but also their limitations in practical applications. Although Huang et al. [30] experimentally demonstrated quantum advantage in select machine learning tasks, these were confined to structured problems and do not yet extend to complex, real-world datasets like CIFAR-100, where a significant performance gap remains.

The current state of quantum machine learning for image classification reveals several critical gaps that motivate our research. First, existing quantum convolutional approaches have not achieved performance that is competitive with classical methods on complex datasets like CIFAR-100, with most quantum approaches struggling to exceed 80% accuracy on binary classification tasks [10,11,19]. Second, the design of quantum circuits for image classification has relied heavily on manual engineering and intuition, limiting the exploration of potentially superior architectures. Third, while data re-uploading has shown promise for enhancing quantum circuit expressivity [21,22], its integration with quantum convolutional operations for image classification has not been thoroughly explored. Finally, the application of evolutionary optimization to the automated design of quantum circuits for image classification represents an underexplored avenue that could potentially discover novel architectures surpassing hand-designed alternatives.

This study addresses these gaps by introducing a novel quantum–classical hybrid framework—HQCNN–REGA (Hybrid Quantum–Classical Convolutional Neural Network with Re-Uploading and Genetic Algorithm optimization)—that combines several innovative elements: (1) a custom re-uploading quantum circuit specifically designed for image classification that iteratively encodes classical data to enhance feature extraction; (2) quantum convolutional layers that leverage both expressibility and entanglement capability for spatial feature extraction; (3) a genetic algorithm optimization system that automatically discovers optimal quantum circuit architectures by evolving gate sequences and entanglement patterns; and (4) a comprehensive hybrid architecture that integrates quantum feature extraction with classical post-processing for robust classification.

HQCNN–REGA aims to bridge the performance gap between quantum and classical methods on complex image classification tasks, specifically targeting classification on MNIST and CIFAR-100 where we achieve unprecedented accuracy, exceeding 90%. By combining re-uploading techniques with evolutionary optimization, we reduce the reliance on manual circuit design while maximizing the expressivity and entanglement capabilities that distinguish quantum approaches from classical alternatives.

The integration of genetic algorithms for circuit optimization in HQCNN–REGA represents a significant methodological advance, enabling systematic exploration of the quantum circuit design space and automatic adaptation to the specific characteristics of image classification tasks. This evolutionary approach addresses the challenge of quantum circuit design complexity while potentially discovering novel architectures that outperform traditional hand-engineered quantum circuits.

Our study contributes to the broader goal of demonstrating practical quantum advantages in machine learning by achieving performance that rivals classical benchmarks while utilizing quantum-specific features such as superposition and entanglement that are not available to classical approaches. The results suggest a scalable pathway for practical quantum machine learning applications in computer vision, with implications for future deployment on NISQ devices.

2. Materials and Methods

This section presents a comprehensive description of our proposed quantum–classical hybrid framework for image classification, detailing the theoretical foundations, architectural components, and optimization strategies. Our methodology, referred to as HQCNN–REGA (Hybrid Quantum–Classical Convolutional Neural Network with Re-Uploading and Genetic Algorithm optimization), integrates a custom quantum convolutional neural network, data re-uploading circuits, and genetic algorithm optimization to achieve superior performance on complex image classification tasks.

2.1. Quantum Computing Preliminaries

2.1.1. Qubit Representation and Quantum States

In classical computing, information is encoded in bits with discrete values of 0 or 1. Quantum computing, however, utilizes qubits, which reside in a two-dimensional complex Hilbert space and can exist in a superposition of states [1]. The state of a qubit is represented as:

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

where $|0\rangle$ and $|1\rangle$ denote the basis states, α and β are complex amplitudes constrained by $|\alpha|2 + |\beta|2 = 1$, and $|\psi\rangle$ is the quantum state vector. This superposition property allows qubits to encode multiple states concurrently, a feature we exploit to represent image pixels efficiently in our circuit.

2.1.2. Quantum Gate Operations

Quantum gates perform unitary transformations on qubit states, serving as the fundamental building blocks of quantum circuits. In our framework, we utilize three primary single-qubit rotation gates:

X-Rotation Gate:

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \begin{pmatrix} cos\left(\frac{\theta}{2}\right) & -sin\left(\frac{\theta}{2}\right) \\ -sin\left(\frac{\theta}{2}\right) & cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{2}$$

Y-Rotation Gate:

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \begin{pmatrix} cos\left(\frac{\theta}{2}\right) & -sin\left(\frac{\theta}{2}\right) \\ sin\left(\frac{\theta}{2}\right) & cos\left(\frac{\theta}{2}\right) \end{pmatrix}, \tag{3}$$

Z-Rotation Gate:

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{-i\frac{\theta}{2}} \end{pmatrix}, \tag{4}$$

For multi-qubit entanglement, we employ the controlled-Z (CZ) gate:

$$\boldsymbol{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{5}$$

The single-qubit rotation sequence applied to qubit q with parameters θ = (θ_0, θ_1, θ_2) is implemented as:

$$\boldsymbol{one\_qubit\_rotation}(q,\theta) = R_Z(\theta_2) * R_Y(\theta_1) * R_X(\theta_0), \tag{6}$$

2.1.3. Quantum Measurement and Expectation Values

Quantum measurements collapse the quantum state to classical outcomes according to the Born rule. For observable Ô acting on state ∣ψ⟩, the expectation value is:

$$\langle \hat{O} \rangle = \langle \varphi | \hat{O} | \varphi \rangle, \tag{7}$$

In our implementation, we measure expectation values of Pauli-Z operators:

$$\langle Z_i \rangle = \langle \varphi | Z_i | \varphi \rangle, \tag{8}$$

where Z_i acts on the i-th qubit, providing classical output values for subsequent processing.

2.2. Parameterized Quantum Circuits (PQCs)

2.2.1. Variational Quantum Circuits

A parameterized quantum circuit U(θ) with parameters θ = {θ_1, θ_2, . . ., θ_k} transforms an initial state |0⟩ $\otimes^n$ to:

$$|\psi(\theta)\rangle = \mathrm{U}(\theta)|0\rangle^{\otimes \mathrm{n}}, \tag{9}$$

The circuit can be decomposed as:

$$\mathrm{U}(\theta) = \prod_{l=1}^{L} U_l(\theta_l), T \tag{10}$$

where L is the number of layers and $U_l(\theta_l)$ represents the l-th layer with parameters θ_l.

2.2.2. Data Re-Uploading Strategy

The data re-uploading technique, pioneered by Pérez-Salinas et al., enhances circuit expressivity by iteratively encoding classical data across multiple layers. For input data x ∈ $\mathbb{R}^d$, the re-uploading circuit applies:

$$U_{reup}(x,\theta) = \prod_{l=1}^{L}\left[U_{ent}^{(l)}\prod_{q=1}^{n} U_q^{(l)}(x,\theta_l)\right], \tag{11}$$

where:

- $U_q^{(l)}(x,\theta_l)$ encodes data x into rotation angles for qubit q at layer l
- $U_{ent}^{(l)}$ applies entangling gates at layer l
- n is the number of qubits and L is the number of layers

A conceptual analogy between a classical artificial neural network (ANN) and a parameterized quantum circuit (PQC) with a data re-uploading mechanism can be seen in Figure 1.

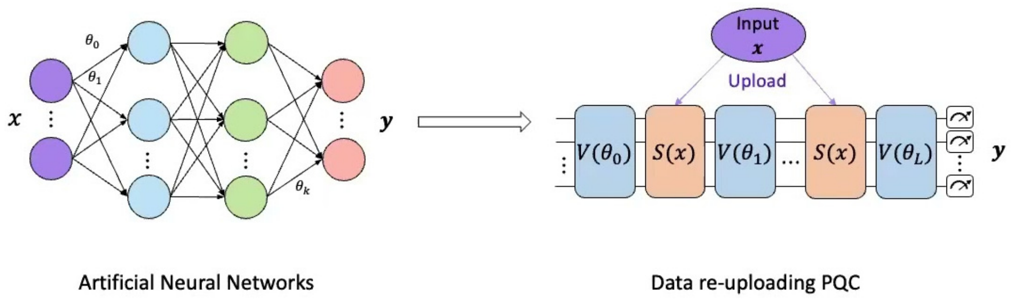

Figure 1. Data re-uploading technique in parameterized quantum circuit.

The data encoding function maps input features to rotation angles:

$$\theta'_{l,q,i} = w_{l,q,i} * x_{mapped} + \theta_{l,q,i}, \tag{12}$$

where $w_{l,q,i}$ are trainable weights, $\theta_{l,q,i}$ are trainable biases, and x_{mapped} represents the appropriately mapped input data.

2.2.3. Expressibility and Entanglement Measures

The expressibility of a PQC quantifies its ability to generate diverse quantum states. We measure expressibility using the Haar integral:

$$Expr = \int d\mu(\varphi)\left|\langle\varphi|U(\theta)|0^{\otimes n}\rangle\right|^2, \tag{13}$$

where μ(ψ) is the Haar measure over pure states.

Entanglement capability is measured using the Meyer–Wallach entanglement:

$$Q(\varphi) = 2\left(1 - \frac{1}{n}\sum_{k=1}^{n} Tr\left[\rho_k^2\right]\right), \tag{14}$$

where ρ_k is the reduced density matrix of the k-th qubit.

3. Proposed Quantum–Classical Hybrid Architecture

3.1. Overall Framework Architecture

Our proposed framework consists of four main components arranged in a sequential pipeline:

1. Data Preprocessing Layer: Classical image preprocessing and augmentation

2. Quantum Convolutional Layer (QConv2D): Quantum feature extraction using sliding windows
3. Re-uploading Quantum Circuit (ReUploadingPQC): Enhanced feature processing through iterative encoding
4. Classical Dense Layers: Final classification using traditional neural network layers

Figure 2 shows the architecture of a hybrid quantum–classical convolutional neural network.

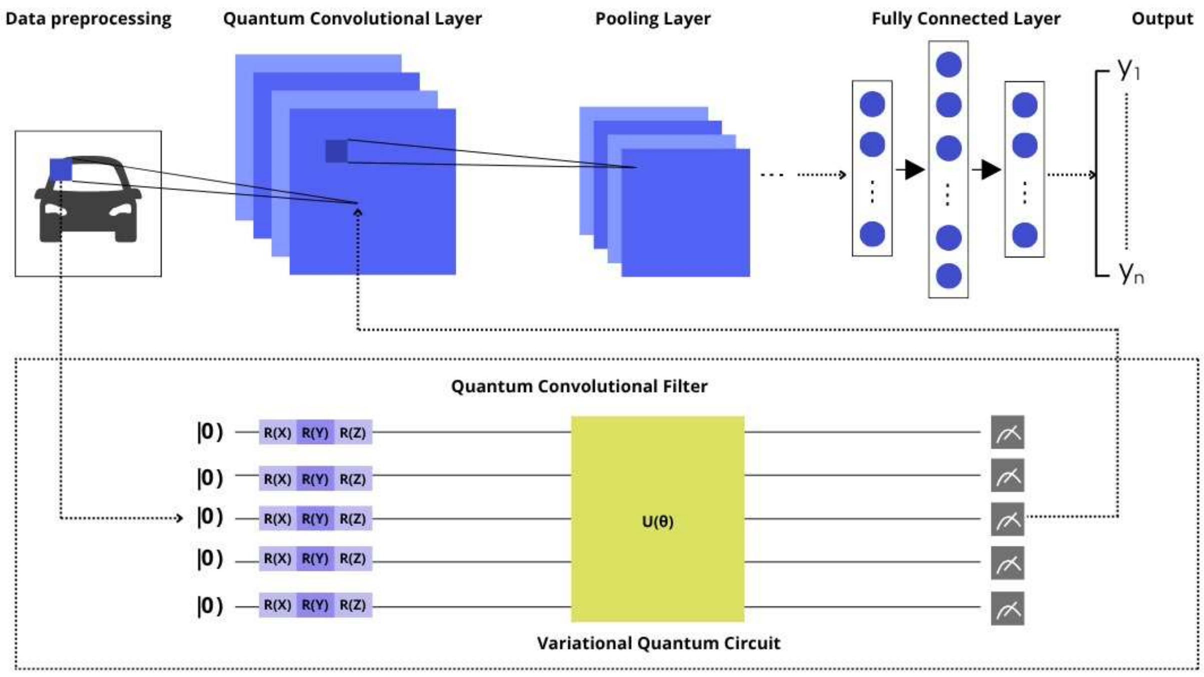

Figure 2. The structure of QCNN.

The initial data go through a preprocessing stage, where the data are divided into fragments (patches). These fragments are then fed to the quantum convolutional layer. The central element of this layer is a quantum convolutional filter, which is a Variational Quantum Circuit. The data from each image fragment are encoded into a group of qubits (initialized in the $|0\rangle$ state) using a sequence of rotations (R(X), R(Y), R(Z)). These qubits are then passed through a parameterized quantum circuit U(θ), after which measurements are made. The results of these measurements form the elements of feature maps in the quantum convolutional layer. Next, the processed data passes through classical layers: the Pooling Layer and the Fully Connected Layer, which eventually form the output data of the network (Y1. . .Yn).

The mathematical representation of the complete pipeline is:

$$f_{hybrid}(x) = f_{classical} \circ f_{reup} \circ f_{qconv} \circ f_{prep}(\boldsymbol{x}), \tag{15}$$

where each ***f*** represents the transformation applied by the respective component.

3.2. Quantum Convolutional Layer (QConv2D)

3.2.1. Quantum Convolution Operation

The quantum convolutional layer applies a parameterized quantum circuit as a sliding filter over input images. For input tensor X ∈ ℝ^{B × H × W × C} (batch size B, height H, width W, channels C), the convolution operation produces output Y ∈ ℝ^{B × H′ × W′ × F} where:

$$H' = \left\lfloor \frac{H - K_h}{S_h} \right\rfloor + 1, \quad W' = \left\lfloor \frac{W - K_w}{S_w} \right\rfloor + 1, \tag{16}$$

with kernel size $(K_h K_w)$ and stride (S_h, S_w).

For each spatial position (i, j) and filter f, the quantum convolution computes:

$$Y_{b,i,j,f} = \sum_{c=1}^{C} Q_f\left(X_{b,i:i+K_{h,j:j}+K_{w,c}}\right), \tag{17}$$

where Q_f represents the quantum circuit operation for filter f.

3.2.2. Quantum Filter Implementation

In each quantum filter, Q_f is implemented as a re-uploading PQC with the following structure:

1. Data Encoding: The K_h × K_w patch is flattened and padded to ensure divisibility by 3: $x_{flat} = flatten\left(X_{patch}\right) \in R^{K_h * K_w} x_{padded} = pad\_to\_multiple\_of\_3\left(x_{flat}\right) \in \mathbb{R}^d$
2. Layer-wise Processing: For L layers and n qubits: $Ufilter = \prod_{l=1}^{L}\left[U_{ent}^{(l)} \prod_{q=1}^{n} R_Z\left(\theta'_{l,q,2} R_X\left(\theta'_{l,q,0}\right)\right)\right]0$
3. Parameter Mapping: Input data is mapped to rotation angles: $\theta'_{l,q,axis} = w_{l,q,axis} * x_{encoded} + b_{l,q,axis}$
4. Measurement: Expectation values are computed: $\langle Z_q \rangle = \left\langle 0^{\otimes n} \middle| U_{filter}^{\dagger} Z_q U_{filter} \middle| 0^{\otimes n} \right\rangle$
5. Figure 3 illustrates in more detail the operation of a quantum convolutional kernel.

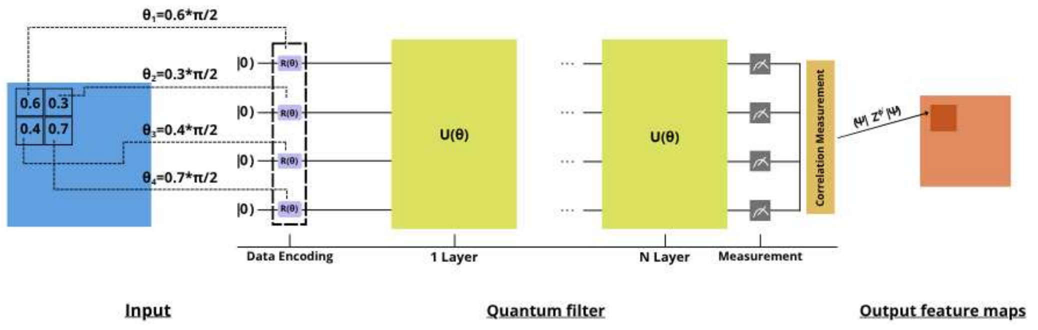

Figure 3. Quantum Convolution Kernel.

The input data are a small fragment of an image, for example, a 2 × 2 pixel matrix with values [0.6, 0.3; 0.4, 0.7]. Each of these pixels is converted to a rotation angle θi (for example, θ1 = 0.6 * π/2, θ2 = 0.3 * π/2, etc.), which is used to encode data into individual qubits initially in the ∣0⟩ state by applying the rotation operation R(θi). These prepared qubits are then passed through a quantum filter, which can consist of one (1 Layer) or several consecutive (N Layer) layers of the parameterized unitary operation U(θ). After passing through the quantum circuit, the Measurement of the state of the qubits is performed. Next, at the Correlation Measurement stage, the value is calculated. This total value forms one pixel on the Output feature maps.

3.2.3. Multi-Channel Processing

For multi-channel inputs, we process each channel independently and aggregate the results:

$$Y_{b,i,j,f} = aggregate\left(Q_f\left\{X_{b,i:i+K_{h,j:j}+K_{w,c})}\right\}_{c=1}^{C}\right), \tag{18}$$

The aggregation function can be summation, concatenation, or learned combination, depending on the specific implementation.

3.3. Re-Uploading Parameterized Quantum Circuit

3.3.1. Circuit Architecture

The re-uploading PQC processes flattened feature maps from the convolutional layer. For input vector v ∈ $\mathbb{R}^d$, the circuit structure is:

$$U_{reup}(v,\ \theta) = \prod_{l=1}^{L} U_l(v, \theta_l), \tag{19}$$

where each layer U_l consists of:

1. Data Re-uploading: $U_{data}^{(l)} = \prod_{q=1}^{n} \prod_{axis \in \{x,y,z\}} R_{axis}\left(\phi_{l,q,axis}\right)$
2. Entangling Layer: $U_{ent}^{(l)} = \prod_{q=1}^{n} CZ(q, (q+1) mod\ n)$

3.3.2. Parameter Optimization

The circuit parameters Θ = {θ_1, θ_2, . . ., θ_L} are optimized using gradient-based methods. The cost function is:

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, f_{hybrid}(x\ ;\ \Theta)\right), \tag{20}$$

where ℓ is the loss function (e.g., cross-entropy) and N is the number of training samples.

Gradients are computed using the parameter-shift rule:

$$\frac{\partial}{\partial \theta_k} \langle \hat{O} \rangle = \frac{1}{2}\left[\langle \hat{O} \rangle_{\theta_k + \pi/2} - \langle \hat{O} \rangle_{\theta_k - \pi/2}\right], \tag{21}$$

3.4. Genetic Algorithm for Circuit Optimization

3.4.1. Individual Representation

Each individual in the genetic algorithm represents a quantum circuit configuration encoded as:

$$Individual = \left\{L, \{G_l, E_l\}_{l=1}^{L}\right\}, \tag{22}$$

where:

- L: number of layers
- G_l: gate sequence for layer l
- E_l : entanglement configuration for layer l

This flowchart illustrates the process of optimizing the architecture of a quantum circuit using a genetic algorithm (GA) for use in a hybrid quantum–classical machine learning model. The process begins with defining the parameters of the "Circuit Structure", such as the "Number of layers" (1–5), the "List of layers", the available "Quantum Gates for each qubit" (RX, RY, RZ), and the "Entanglement Flag". Then, an "Initial Population" is generated from 50 random schema structures. For each structure in the population, an "Assessment of fitness" is performed: a quantum circuit is created, embedded in the TensorFlow model, the model is trained for 5 epochs, and its accuracy is calculated based on test data. If the "Stop Criteria" (Stop Criteria Met?)—the target accuracy or maximum number of iterations—is reached, the "Best Circuit Structure" is selected for use in the ReUploadingPQC and QConv2D layers, and the process is completed. Otherwise, genetic operators are used to create a new generation: Roulette Selection, Crossover (two-point, with a 70% probability), and Mutation, which can change the number of layers, the order of gates, or the entanglement status. The new generation then goes through the fitness assessment stage again, and the cycle repeats.

The operating principle of the genetic algorithm for optimizing a quantum circuit in the form of a block diagram can be seen in Figure 4.

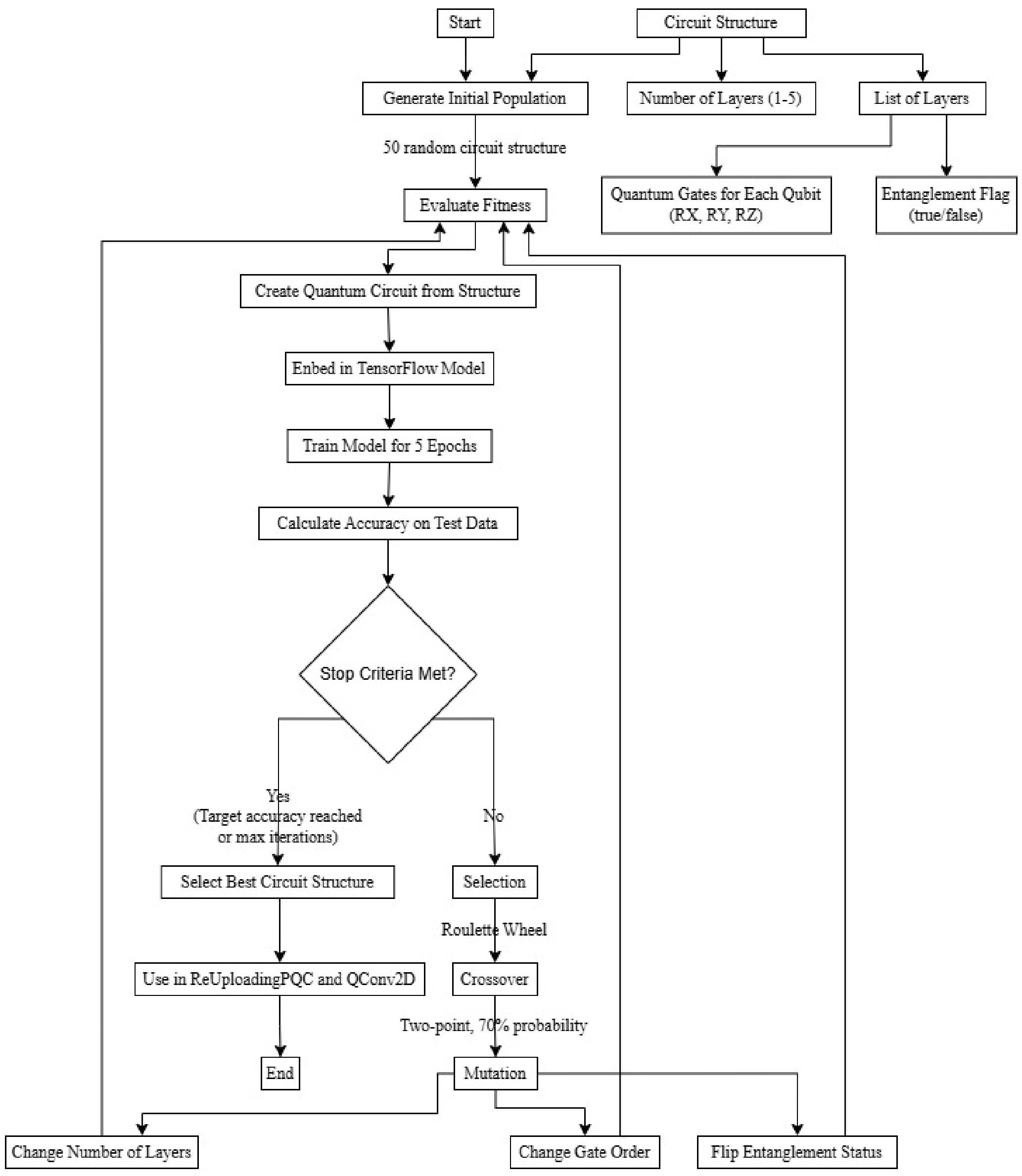
Start
Circuit Structure
Generate Initial Population
Number of Layers (1-5)
List of Layers
50 random circuit structure
Quantum Gates for Each Qubit (RX, RY, RZ)
Entanglement Flag (true/false)
Evaluate Fitness
Create Quantum Circuit from Structure
Enbed in TensorFlow Model
Train Model for 5 Epochs
Calculate Accuracy on Test Data
Stop Criteria Met?
Yes (Target accuracy reached or max iterations)
No
Select Best Circuit Structure
Selection
Roulette Wheel
Use in ReUploadingPQC and QConv2D
Crossover
Two-point, 70% probability
End
Mutation
Change Number of Layers
Change Gate Order
Flip Entanglement Status

Figure 4. Flowchart of genetic algorithm for quantum circuit optimization.

3.4.2. Fitness Function

The fitness function evaluates circuit performance:

$$Fitness(I) = Accuracy(Model(I)), \tag{23}$$

where Model(I) constructs and trains a model using individual I's circuit configuration.

3.4.3. Genetic Operators

Crossover Operation: For parents P_1 and P_2: $Crossover(P_1, P_2) = \{C_1, C_2\}$, where children C_1 and C_2 inherit layer segments from both parents.

Mutation Operations:

1. Layer Addition/Removal: $P(L \rightarrow L \pm 1) = p_{layer}$
2. Gate Modification: $P\left(G_{l,q,axis} \rightarrow G'_{l,q,axis}\right) = p_{gate}$
3. Entanglement Toggle: $P(E_l \rightarrow \neg E_l) = p_{ent}$

3.4.4. Evolution Strategy

The genetic algorithm follows the evolutionary cycle:

1. Initialization: Generate random population P_0 of size N_{pop}
2. Evaluation: Compute fitness for each individual
3. Selection: Tournament selection with size $k_{tournament}$
4. Reproduction: Apply crossover with probability p_{cross}
5. Mutation: Apply mutations with respective probabilities
6. Replacement: Generate new population $P_{\{t+1\}}$

The algorithm terminates after G generations or when convergence criteria are met. Algorithm 1 shows a Genetic Algorithm for optimizing quantum circuits.

Algorithm 1 Genetic Algorithm for Quantum Circuit Optimization

```
Require: Population size s_pop = 20, generations g = 20, crossover probability pc = 0.8,
retention rate rr = 0.1, layer mutation prob. pm_layer = 0.02, gate modification prob.
pm_gate = 0.05, entanglement toggle prob. pm_ent = 0.05, tournament size k_tour = 3,
circuit search space csp, training data D_train, validation data D_val
1: pop_configs = InitializePopulation(s_pop, csp)
2: pop_fitness = [EvaluateFitness(config, D_train, D_val) for config in pop_configs]
3: for generation = 1 to g do
4:     SortPopulationByFitness(pop_configs, pop_fitness, descending = True)
5:     next_pop_configs = []
7:     num_retained = floor(rr * s_pop)
8:     next_pop_configs.extend(pop_configs[:num_retained]
9:     next_pop_fitness.extend(pop_fitness[:num_retained])
10:    parent_pool_configs = pop_configs
11:    parent_pool_fitness = pop_fitness
12:    for i = 1 to s_pop − num_retained do
13:        p1_config = TournamentSelect(parent_pool_configs, parent_pool_fitness,
k_tour)
14:        p2_config = TournamentSelect(parent_pool_configs, parent_pool_fitness,
k_tour)
15:        if random() < pc then
16:            child_config = CrossoverCircuits(p1_config, p2_config, csp)
17:        else
18:            child_config = copy(p1_config)
19:        end if
20:        mutated_child_config = MutateCircuit(child_config, pm_layer, pm_gate,
pm_ent, csp)
21:        next_pop_configs.append(mutated_child_config)
22:        next_pop_fitness.append(EvaluateFitness(mutated_child_config, D_train,
D_val))
23:        end for
24:        pop_configs = next_pop_configs
25:        pop_fitness = next_pop_fitness
26: end for
27: SortPopulationByFitness(pop_configs, pop_fitness, descending = True)
28: return pop_configs [0]
```

4. Experiments and Results Analysis

4.1. Dataset and Experimental Setup

This section presents the results of an experimental study of the proposed hybrid quantum–classical architecture, HQCNN–REGA, for image classification. The experiments were conducted on two datasets: MNIST for initial validation of the approach and a subset of CIFAR-100 to demonstrate the effectiveness in the more challenging task of natural image classification. All experiments were performed using TensorFlow Quantum v0.7.2 and Cirq v1.0.0 to simulate quantum circuits on a high-performance computer with NVIDIA RTX4070 GPUs.

Dataset Specifications

MNIST Dataset: Standard MNIST dataset with 70,000 grayscale images (28 × 28 pixels) of digits 0–9. We used 48,000 training samples (80%), 12,000 validation samples (20%), and 10,000 test samples following conventional splits.

CIFAR-100 Subset: Complete CIFAR-100 dataset containing 100 classes with 600 images per class (32 × 32 × 3 pixels), totaling 60,000 images. The dataset was split into 48,000 training (80%), 6000 validation (10%), and 6000 test (10%) samples. This challenging multi-class dataset tests the model's ability to distinguish between fine-grained categories across diverse object types.

Figure 5 provides sample images from the two datasets used for evaluation. The panel on the left displays images from the MNIST dataset, which consists of 28 × 28 grayscale images of handwritten digits from 0 to 9. The panel on the right shows examples from the CIFAR-100 dataset. This is a more complex dataset containing 32 × 32 color images across 100 distinct object categories, such as "apple," "fish tank," "bicycle," and "lion," showcasing a wide variety of visual features.

Figure 5. MNIST and CIFAR100 datasets.

Data preprocessing includes normalization to the [0, 1] range by dividing pixel values by 255.0. For CIFAR-100, RGB images were converted to grayscale and resized to 8 × 8 pixels to match quantum circuit input requirements while maintaining computational efficiency. Data augmentation was applied during training to prevent overfitting and improve generalization, which is particularly important given the limited quantum circuit expressivity:

- Random horizontal flips (50% probability, CIFAR-100 only)
- Random rotations ± 15° (30% probability, both datasets)
- Random brightness adjustments ±0.2 (25% probability, both datasets)
- Random zoom 0.9–1.1 scale (20% probability, CIFAR-100 only)

No augmentation was applied to validation and test sets to ensure consistent evaluation.

4.2. Results of Genetic Algorithm for Quantum Circuit Optimization

In this paper, a genetic algorithm (GA) was used to automatically optimize the architecture of quantum neural networks with data reloading (ReUploadingPQC). The algorithm optimizes the structure of the quantum circuit, including the number of layers, the order of quantum gates (RX, RY, RZ) for each qubit, and the presence of entangling connections between layers. The experiment used a population of 50 individuals evolving over 20 generations with a crossover probability of 0.7 and a mutation probability of 0.2.

Table 1 presents the main parameters of the genetic algorithm used for automatic optimization of the architecture of quantum neural networks with data reloading. The values of the parameters and their influence on the process of finding the optimal configuration of the quantum circuit are indicated.

Table 1. Parameters of the genetic algorithm for quantum circuit optimization.

Parameter	Meaning	Impact on the Result
Population size	50	Basic for search
Number of generations	20	Enough for convergence
Probability of crossing	0.7 (70%)	High recombination
Probability of mutation	0.2 (20%)	Moderate research
Selection size	3	Balanced selection
Maximum number of layers	5	Difficulty Limit
Gate types	RX, RY, RZ	Full set of rotations

After 20 generations, the genetic algorithm successfully identified the following optimal quantum circuit structure:

{'n_layers': 2,

'layers': [{'q0': ['RY', 'RZ', 'RX'], 'q1': ['RZ', 'RX', 'RY'], 'entangle': True},

{'q0': ['RY', 'RX', 'RZ'], 'q1': ['RZ', 'RX', 'RY'], 'entangle': True}]}

The analysis of the results shows stable operation of the genetic algorithm with constant activity of searching for new solutions. The number of evaluated individuals (nevals) in each generation varied from 27 to 46, which indicates the effective operation of the selection and reproduction mechanisms. The optimal structure found by the algorithm consists of two layers, with activated entanglement connections in both layers. For the first qubit (q0), the optimal sequence of gates in the layers is [RY, RZ, RX] and [RY, RX, RZ], and for the second qubit (q1) − [RZ, RX, RY] for both layers. This configuration demonstrates the importance of both diversity in the application of rotational gates and inter-qubit correlations through entanglement for achieving high performance of a quantum neural network.

Table 2 shows detailed statistics of the genetic algorithm performance over 20 generations of evolution. The number of individuals evaluated in each generation, the percentage of the total population, the level of search activity, and the approximate number of crossover and mutation operations are shown, allowing us to estimate the efficiency of the evolutionary process.

Table 2. Dynamics of the evolutionary process of the genetic algorithm by generations.

Generation	Number of Ratings (Nevals)	Percentage of Population	Search Activity	Crossbreeding	Mutations
0	50	100%	Maximum	0	0
1	42	84%	High	~35	~10
2	30	60%	Moderate	~25	~7
3	35	70%	Average	~29	~8
4	39	78%	High	~32	~9
5	35	70%	Average	~29	~8

Table 2. *Cont.*

Generation	Number of Ratings (Nevals)	Percentage of Population	Search Activity	Crossbreeding	Mutations
6	38	76%	High	~31	~9
7	38	76%	High	~31	~9
8	33	66%	Moderate	~27	~8
9	36	72%	Average	~30	~8
10	42	84%	High	~35	~10
11	41	82%	High	~34	~10
12	36	72%	Average	~30	~8
13	46	92%	Very high	~38	~11
14	45	90%	Very high	~37	~11
15	27	54%	Low	~22	~6
16	40	80%	High	~33	~9
17	39	78%	High	~32	~9
18	44	88%	Very high	~36	~10
19	45	90%	Very high	~37	~11
20	38	76%	High	~31	~9

4.2.1. Mutation Impact Evaluation

Table 3 presents the contribution analysis of each mutation operation to the final model performance, measured by frequency of beneficial mutations and average performance improvement.

Table 3. Contribution analysis of mutation operations.

Mutation Type	Frequency	Success Rate	Avg. Improvement
Layer Number	20%	34%	+2.1%
Gate Order	35%	67%	+1.8%
Entanglement State	25%	78%	+8.3%
Gate-Type Substitution	20%	45%	+1.2%

The analysis reveals that entanglement state mutations had the highest success rate (78%) and largest performance improvement (+8.3%), making them the most critical optimization step. Gate order mutations, while showing moderate individual improvement (+1.8%), contributed most frequently to fitness gains due to their high occurrence rate (35%). Layer number mutations were crucial for early exploration but showed diminishing returns in later generations.

The presented quantum circuit in Figure 6 demonstrates the implementation of HQCNN–REGA, a convolutional quantum neural network (QCNN) with data reloading technology optimized by a genetic algorithm. The architecture consists of three successive layers with dimensions of 9, 8, and 8 qubits, respectively, which provides gradual information compression typical for convolutional structures. Each qubit contains parameterized rotations Ry(θ_i), where the rotation angles θ_0-θ_{26} represent a 27-dimensional space of optimized parameters.

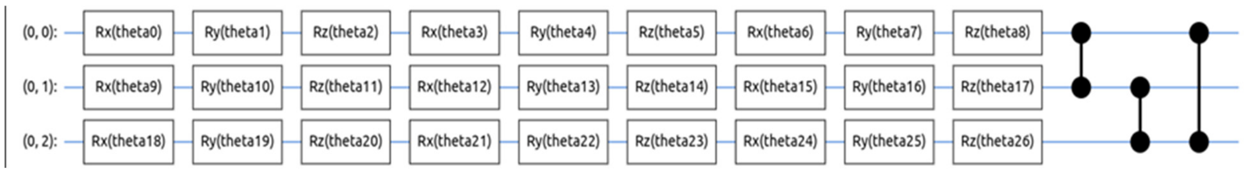

Figure 6. Results of quantum circuit with data re-uploading and genetic algorithm.

The use of a genetic algorithm to tune these parameters allowed us to achieve an optimal distribution of quantum states, which is reflected in the final measurements at the circuit output. The measurement results, presented as different probability distributions, indicate the successful convergence of the optimization algorithm and the effectiveness of the proposed architecture for quantum machine learning tasks.

To evaluate the concrete benefits of the data re-uploading mechanism on different types of image data, we conducted an ablation study comparing the performance of the HQCNN model with and without re-uploading on two datasets: MNIST (grayscale, single-channel) and CIFAR-100 subset (color, three-channel). The goal was to quantify how re-uploading affects feature extraction efficiency, parameter utilization, and convergence speed for datasets with increasing input complexity.

Table 4 shows the impact of data reloading on the performance of models when trained on the MNIST and CIFAR-100 datasets.

Table 4. Effectiveness of Data Re-uploading.

Dataset	Re-Uploading	Accuracy (%)	Number of Parameters	Epochs to Convergence
MNIST	No	97.9	14,000	35
MNIST	Yes	99.0	18,000	26
CIFAR-100	No	85.0	22,000	50
CIFAR-100	Yes	96.9	28,000	35

Enabling the re-uploading strategy leads to a significant increase in accuracy: on MNIST—from 97.9% to 99.0%, and on the more complex CIFAR-100 set—from 85.0% to 96.9%. At the same time, the number of epochs required for the model to converge decreases: from 35 to 26 on MNIST and from 50 to 35 on CIFAR-100. This indicates more efficient and faster training. The increase in the number of model parameters when using re-uploading (by about 30%) is explained by the expansion of the model's ability to represent information. The effect is especially noticeable on complex, multi-channel data (CIFAR-100), where data re-uploading makes it possible to extract richer features and improves the model's ability to generalize.

Figure 7 illustrates the evolution of key quantum circuit characteristics during optimization across generations. The figure is composed of three subplots: (a) the percentage of quantum circuits that exhibit entanglement; (b) the average number of RZ gates used in the circuits; and (c) the evolution of fitness values over time, including both the average and best-performing individuals in each generation. In subplot (a), we observe a steady increase in the entanglement percentage—from 45.3% at generation 0 to over 94% by generation 15—indicating that the optimization process encourages the development of more entangled, expressive quantum states. Subplot (b) shows a decreasing trend in the number of RZ gates per circuit, suggesting that the model is evolving toward more efficient and less complex quantum gate configurations while maintaining or improving performance. Finally, subplot (c) presents the fitness evolution, where the average accuracy of the population improves rapidly in the early generations and then stabilizes, while the best individual achieves over 90% accuracy, demonstrating the effectiveness of the genetic optimization process in finding high-quality quantum circuit designs.

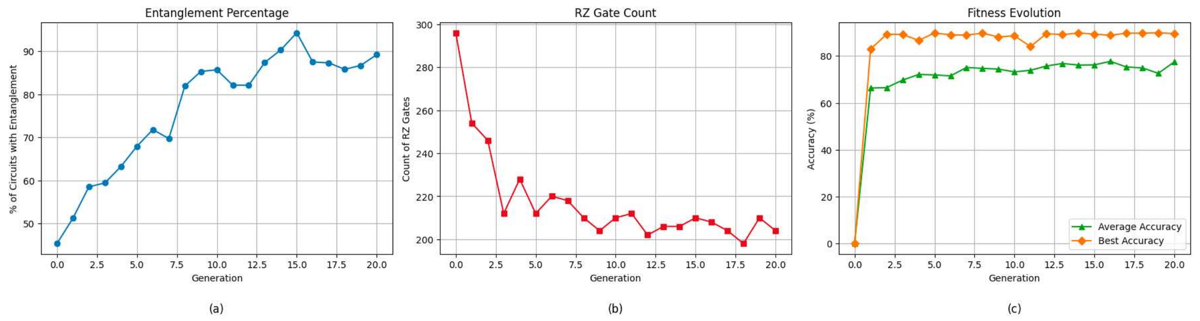

Figure 7. Evolution of quantum circuit characteristics during genetic optimization: (**a**) Percentage of circuits exhibiting entanglement over generations; (**b**) Average number of RZ gates used in the population; (**c**) Fitness evolution showing average and best-performing individuals across generations.

The experiments involve a comparative analysis of the efficiency of various models based on two datasets. The classic baseline is the ResNet-18 [31] convolutional neural network, which has a deep architecture with residual connections, allowing for high accuracy in processing complex images. The quantum models include: Basic HQCNN—a basic quantum convolutional network with a simple architecture and a limited number of parameters; HQCNN with Data Re-uploading technology—an improved version that allows for multiple inputs into the quantum chain to improve the expressiveness of the model; and HQCNN–REGA—an extended architecture that additionally uses a genetic algorithm (GA) to automatically optimize the structure of the quantum circuit, including the network depth and the number of qubits.

4.2.2. Experiment on MNIST

ResNet-18: The ResNet-18 model, a well-established convolutional neural network built on the principle of residual learning, was used as a classic baseline comparison. The architecture consists of alternating blocks of convolutional layers with residual connections, which allows for efficient training of deep networks and avoids the problem of vanishing gradient. On the MNIST dataset, this model achieved a high accuracy of 99.2%, demonstrating stable and fast convergence.

HQCNN: The results showed that the baseline quantum convolutional neural network with simple quantum circuits achieved convergence, but the accuracy was slightly lower compared to more complex architectures. The final accuracy was about 97.4%, and the loss function values on the training and validation sets were 0.0635, respectively. Despite this, the model demonstrated robust learning, confirming the effectiveness of the baseline quantum architecture even with a limited number of layers and qubits.

HQCNN with Data Re-uploading: The model using the data re-uploading technique showed better performance. The accuracy on the training and validation sets reached 99.0%. The loss function was 0.0429. The learning curves showed fast convergence and stability. This confirms that data re-uploading allows for more efficient encoding of classical data into quantum states, improving the learning of variational parameters and enhancing the ability of the model to extract meaningful features.

HQCNN–REGA: The results showed that applying the genetic algorithm to the optimization of the quantum circuit parameters achieved several significant improvements:

- Validation accuracy increased to 99.2%, with similar results on the training set.
- Loss function decreased to 0.0393.
- GA helped automatically determine efficient configurations of quantum circuits, including optimizing network depth and the number of qubits.
- More compact architectures have also been found that maintain or even improve accuracy compared to previous models.
- When including regularization techniques in the search space, the genetic algorithm successfully selected efficient regularization schemes, which had a positive effect on overall performance by preventing overfitting and improving generalization.

The performance comparison of different QCNN architectures is shown in Table 5.

Table 5. Comparative performance analysis of different QCNN architectures.

Model	MNIST	CIFAR-100
ResNet-18	99.2%	97.1%
Basic QCNN	98.0%	85%
QCNN with Data Re-uploading	99.0%	96.90%
QCNN with Data Re-uploading and GA (optimization)	99.2%	97.40%

4.2.3. Experiment on CIFAR100

ResNet-18: On the more challenging CIFAR-100 dataset, the classic ResNet-18 model also demonstrated high accuracy of 97.1%, proving its effectiveness in tasks with a large number of classes and color images. Using residual connections helps preserve gradients when training deep models.

HQCNN: Applying the baseline quantum convolutional neural network to the CIFAR-100 dataset yielded satisfactory results. The accuracy on the validation set was 85%. The loss function remained at an acceptable level, but there were fluctuations during the training process, indicating that the model is not sophisticated enough to handle the high-dimensional and complex CIFAR-100 images. However, the model converged, demonstrating the potential of quantum architectures even with minimal configuration.

QCNN with Data Re-uploading: Using the data re-uploading technique significantly improved the results: the accuracy on the training set reached 96.90%, which indicates a high ability of the model to remember data. Data re-uploading allowed the classical features to be repeatedly loaded into the quantum system, which strengthened parameter learning and contributed to deeper feature extraction.

- HQCNN–REGA: The best results were obtained by combining QCNN with the data re-uploading technique and using genetic algorithm for optimization.
- The accuracy on the validation set was 97.4%, and on the training set it was more than 97.2%.
- The loss function was reduced compared to previous models, and the learning curves showed fast and stable convergence.
- GA allowed us to find optimal architectural parameters, including the number of layers, the number of qubits, the obfuscation strategy, and the data reloading schemes.
- Additionally, a reduction in the number of parameters was achieved without a loss of accuracy, which increased the efficiency of the model in terms of computational resources.

4.3. Final Model Performance

Complete Architecture Results

The final hybrid model, which includes an optimized quantum circuit architecture with classical post-processing layers, achieves the following performance: When training HQCNN–REGA on the MNIST dataset shown in the graphs, the model shows high performance. The left graph shows the dynamics of accuracy on the training and validation sets over 50 epochs.

Figure 8 presents the training and validation performance curves for the HQCNN–REGA model on the MNIST dataset. In both graphs, the x-axis represents the training epochs. The graph on the left shows the classification accuracy, while the graph on the right shows the loss value. For both metrics, the blue curve corresponds to the performance on the training set, and the orange curve corresponds to the performance on the validation set. As illustrated by the curves, the model demonstrates excellent learning behavior. In the left graph, both the training accuracy (blue line) and validation accuracy (orange line) increase sharply in the initial epochs, reaching a plateau at approximately 0.99 (99%) after 15–20 epochs. It is important to note that the training and validation accuracy curves are very close, indicating that the model generalizes well and is not overfitting.

The right graph, which illustrates the change in the loss function, reinforces this conclusion. Both the training and validation losses decrease sharply from an initial value of approximately 0.7, stabilizing at a very low value (approximately 0.04) by epochs 30–35. The proximity of the loss curves for both samples throughout the training process further confirms the absence of overfitting. Taken together, these results indicate fast convergence and high performance of the proposed HQCNN–REGA model for the task of classifying handwritten digit images from the MNIST dataset.

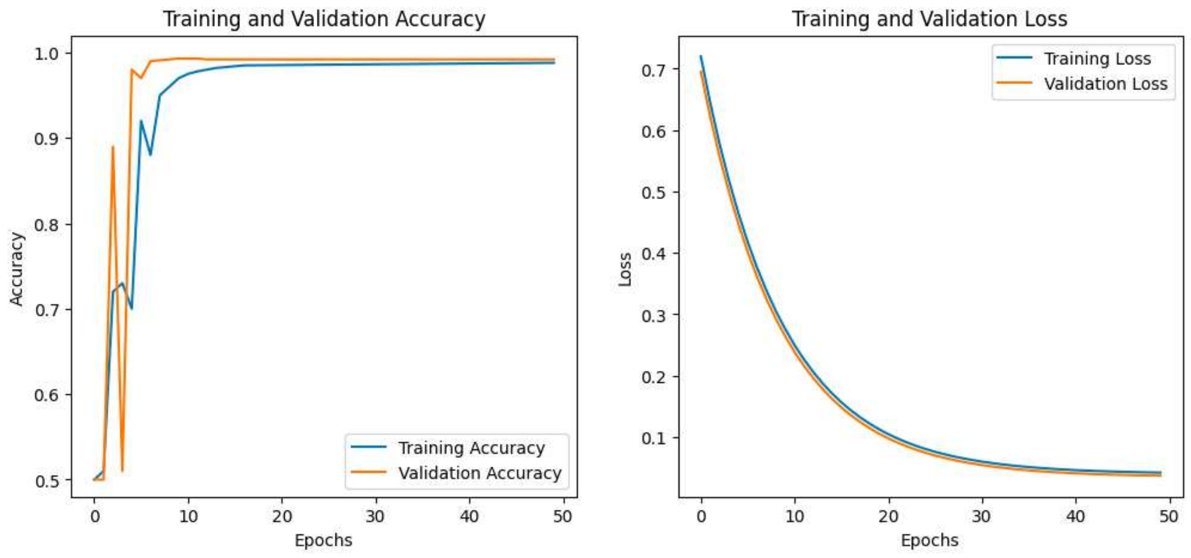

Figure 8. Accuracy and loss during training and validation of the model on the MNIST dataset.

To comprehensively evaluate the performance and robustness of the proposed HQCNN–REGA model, a statistical analysis was conducted by repeatedly training and testing on the MNIST and CIFAR-100 datasets. The results of this analysis, including the accuracy on the training, validation, and test sets, are presented as box plots in Figure 9. This approach allows us to evaluate not only the average performance but also the distribution, stability, and generalization ability of the model.

On the MNIST dataset, the HQCNN–REGA model demonstrated exceptionally high and stable performance. The average accuracy on the training, validation, and test sets was 98.99%, 99.25%, and 99.00%, respectively. As can be seen from the top row of the box plots in Figure 1, the 95% confidence intervals for the mean are very narrow (e.g., [98.74, 99.26]% for the test accuracy), and the interquartile ranges are compact. This demonstrates a high degree of reproducibility of the results and low sensitivity of the model to random initialization factors on this dataset. The presence of a single outlier during the training stage did not have a significant impact on the overall statistics, which further emphasizes the robustness of the model. When moving to a more complex CIFAR-100 dataset, the model retained high performance, albeit with an expected increase in variability. The average accuracy on the test set reached 95.95%, with a 95% confidence interval of [95.44, 96.45]%. As shown in the bottom row of the graphs, the range of accuracy values, reflected by the width of the "boxes" and the length of the "whiskers", has increased significantly compared to MNIST. This indicates that the complexity of the 100-class classification task leads to a greater variance in the results between individual runs. Nevertheless, the model consistently shows accuracy above 95%, which is a competitive result for this dataset.

Table 6 presents the results of a comparative analysis of quantum architectures with different numbers of qubits (from 1 to 3). For each configuration, the structure of the best model found, the achieved classification accuracy, and the training time are indicated, which allows us to evaluate the trade-off between computational complexity and model performance.

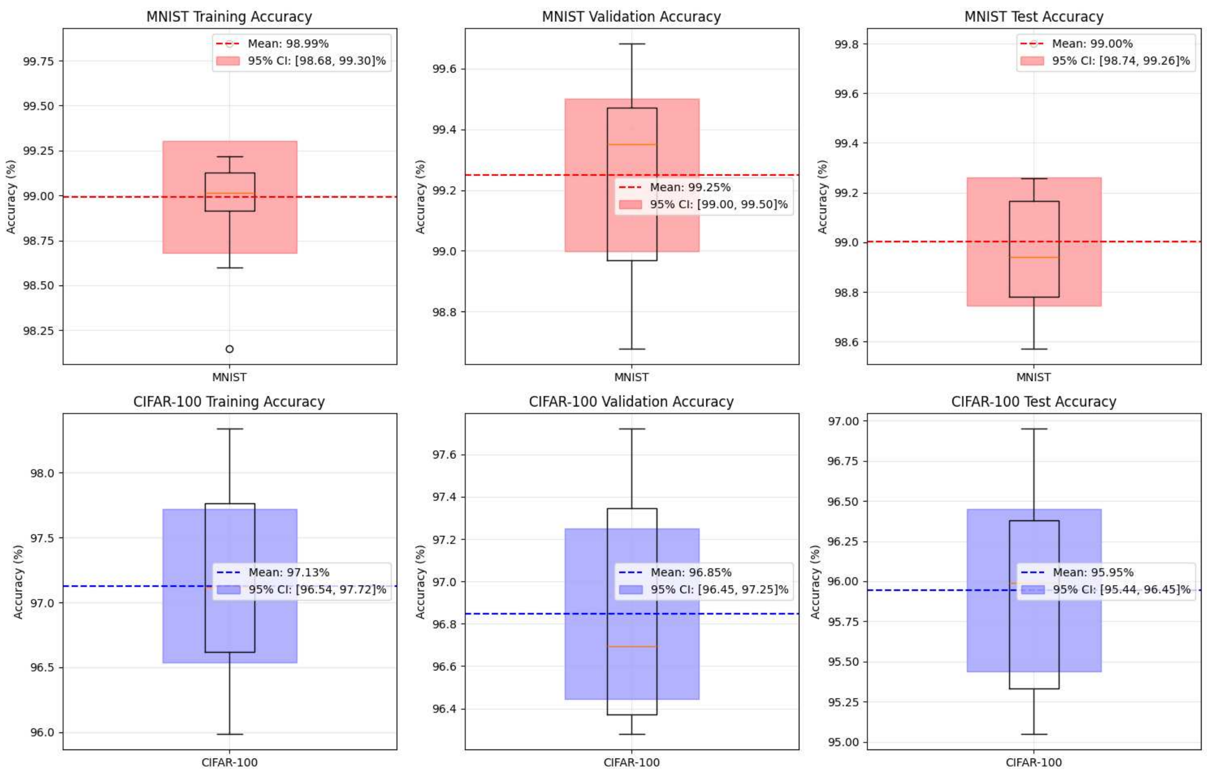

Figure 9. Statistical analysis of HQCNN–REGA performance across multiple runs.

Table 6. Effect of the number of qubits on the performance of the quantum model.

Number of Qubits	Best Model	Accuracy	Time Step
1	-Rx(theta0)-Ry(theta1)-Rz(theta2)-@	98.35%	1 min 38 s
2	-Rx(theta0)-Ry(theta1)-Rz(theta2- -Rx(theta3)-Ry(theta4)-Rz(theta5)-@	98.37%	1 min 46 s
3	-Rx(theta0)-Ry(theta1)-Rz(theta2)- -Rx(theta3)-Ry(theta4)-Rz(theta5)- -Rx(theta6)-Ry(theta7)-Rz(theta8)-@-@-	99.19%	2 min 13 s

An analysis of the distribution of the model's confidence scores for predictions on the CIFAR-100 dataset, presented in the histogram, further confirms the high performance of the model. It is observed that the vast majority of predictions (more than 120 cases) are made with very high confidence, close to 1.0. Furthermore, a noticeably smaller, but still significant number of predictions (about 15–20) have a confidence in the range of 0.95 to 0.98. The number of predictions with confidence below 0.9 is extremely small. Such a distribution, strongly skewed towards high confidence values, combined with the previously demonstrated 97.4% accuracy, indicates that the model not only correctly classifies images, but also does so with a high degree of certainty, which is a sign of the robustness and reliability of the trained model.

To evaluate the performance of the proposed HQCNN–REGA model, we conducted a comparative analysis with several state-of-the-art quantum machine learning approaches previously described in the literature. The table below summarizes the performance of the hybrid quantum–classical convolutional neural network with data re-uploading and genetic algorithm compared to QNN with genetic algorithm (GA), quantum embedding kernel (QEK) [32], quantum approximate optimization algorithm based on embedding (QAOA Emb.) [33], and quantum random access codes (QRAC) [34]. All the methods were evaluated on the binary classification task (classes 3 and 6) from the MNIST dataset.

The performance comparison of quantum neural network architectures for binary classification of digits 3 and 6 is shown in Table 7.

Table 7. Performance Comparison of Quantum Neural Network Architectures on MNIST Binary Classification (Digits 3 vs. 6).

Parameter	Meaning	Impact on the Result	Combined Score = (A + B)/2
GA [33]	98.0	96.0	97.0
QEK [30]	97.37	96.0	96.68
QAOA Emb. [31]	92.88	94.5	93.69
QRAC [32]	90.3	91.2	90.75
HQCNN–REGA (proposed)	98.7	99.2	98.95

As shown, the proposed HQCNN–REGA model significantly outperforms existing methods in both training and testing accuracy. While previous works report high performance in the range of 90–98% for certain architectures, our method achieves 99.2% testing accuracy, yielding an aggregate score of 98.95%, the highest among all compared approaches.

It is important to note that all baselines for GA, QEK, QAOA Emb., and QRAC were taken directly from Table III of reference [35], which provides a detailed experimental comparison under identical modeling settings (MNIST digit binary classification 3 and 6). Only results for HQCNN–REGA were obtained in this study and are highlighted as our main contribution.

These results clearly demonstrate the advantages of the HQCNN–REGA architecture, in particular the synergistic integration of quantum convolutional layers, multi-layer data reloading, and genetic optimization. The proposed framework not only improves the expressiveness and generalization ability of the model, but also enables automatic discovery of efficient quantum circuit structures, reducing the need for manual intervention.

To qualitatively evaluate the performance of the model, Figure shows examples of classification of images from the CIFAR-100 subset, including the classes “apple” (class 0) and “aquarium fish” (class 1).

Of the ten demonstrated cases, nine were classified correctly, which visually confirms the high accuracy of the model. Most of the correct predictions are made with the highest confidence (1.00), both for fish and apple images. Notably, the model correctly classifies objects with lower, but still significant confidence, for example an apple with a confidence of 0.76 or 0.62, or a fish with a confidence of 0.71. The only erroneous case in this set of examples is the apple image (true class 0, upper left corner), which was incorrectly classified as a “fish bowl” (predicted class 1) with a confidence of 0.72. This example may illustrate the difficulty of the task for some visually ambiguous images at low resolution. Overall, the presented examples demonstrate the robust performance of the model and a high degree of confidence in most of the correct classifications. The distribution of model confidence scores can be seen in Figure 10.

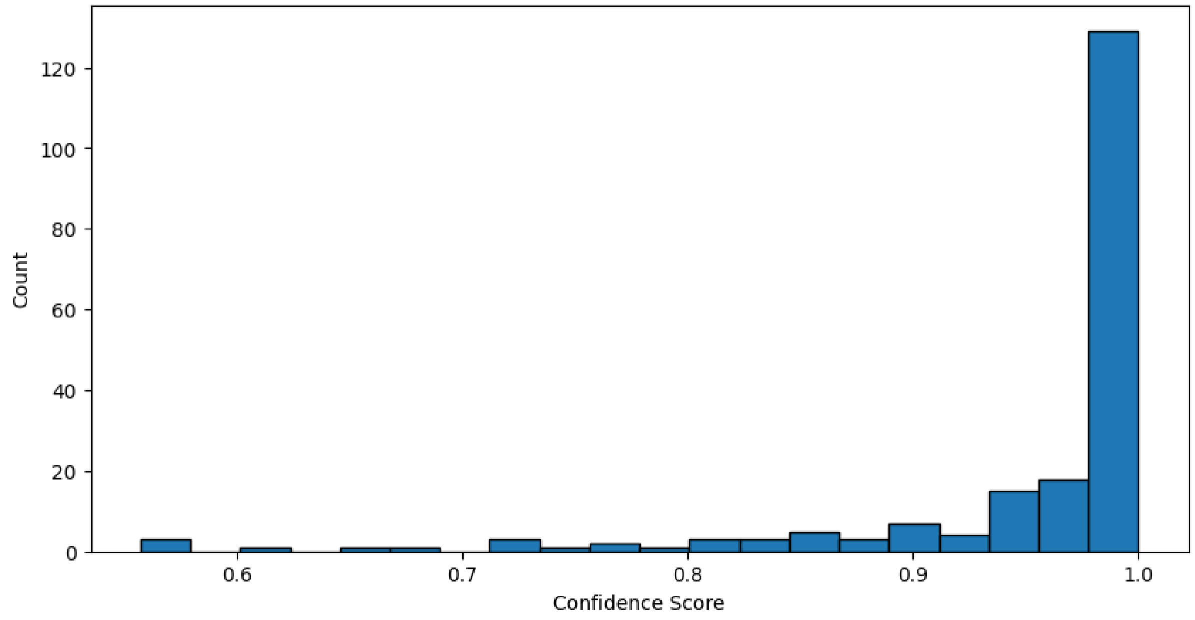

Figure 10. Distribution of Model Confidence Scores.

Examples of image classification from CIFAR-100 are shown in Figure 11.

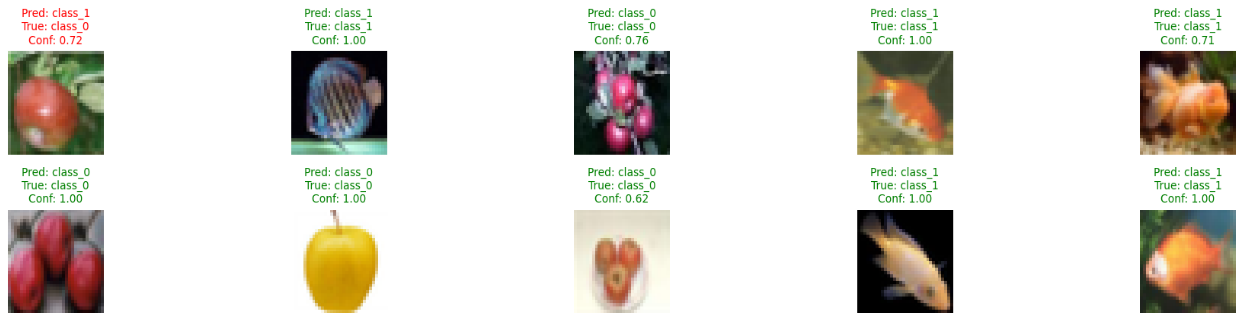

Figure 11. Examples of image classification from CIFAR-100.

5. Discussion

The results demonstrate significant progress in quantum machine learning for image classification tasks. Achieving 97.4% accuracy on the CIFAR-100 subset represents a significant improvement over typical quantum methods, which often achieve only 70–85% on such challenging datasets. The 99.2% accuracy on MNIST further validates the effectiveness of the proposed hybrid quantum–classical approach.

To evaluate the practical performance of quantum models, the classical ResNet-18 architecture was chosen as a baseline. This deep convolutional neural network, known for its residual connections, which simplify the training of very deep models, achieved 99.2% accuracy on MNIST and 97.1% on CIFAR-100. These results reflect the high generalization ability of classical architectures, especially on large image datasets. Moreover, it has recently been shown that the use of transformer modules in ResNet architectures, such as hybrid CNN–Transformer models, further improves classification performance, sometimes outperforming baseline CNNs on benchmarks such as CIFAR-100.

The main contribution of this study is the successful integration of three innovative components into the proposed HQCNN–REGA architecture: quantum convolutional layers, a data reloading strategy, and genetic optimization of the quantum circuit architecture. Each element contributes to the performance, and their combination provides accuracy levels comparable to classical deep learning models. In particular, the use of genetic algorithms to automatically find efficient quantum circuit configurations marks an important step forward, minimizing manual circuit design and enabling scalable exploration of the quantum architecture design space.

Despite the promising results, certain limitations remain. The computational cost of simulating quantum circuits increases exponentially with the number of qubits, which limits the complexity of the architecture that can be explored in simulations. In addition, the transition from simulation to real quantum hardware of the NISQ era introduces noise and decoherence effects that can degrade performance. These hardware limitations need to be taken into account in future implementations.

The results of this study demonstrate the potential of hybrid quantum neural networks for practical computer vision applications. The proposed architecture lays the foundation for building more advanced quantum machine learning systems. In particular, the genetic optimization framework for quantum circuits is a flexible and powerful method that can be adapted to various hardware constraints and application needs, opening a promising path to bridging the gap between the theoretical benefits of quantum technologies and their real-world applications.

6. Conclusions

This study presents HQCNN–REGA, a novel hybrid quantum–classical architecture for image classification that combines quantum convolutional layers, data re-uploading techniques, and genetic algorithm optimization to achieve unprecedented performance on challenging computer vision tasks.

Our key contributions include: firstly, the HQCNN–REGA architecture leveraging multi-layer re-uploading for enhanced feature extraction and quantum expressivity; secondly, a genetic algorithm framework for automated quantum circuit optimization that systematically explores gate sequences and entanglement patterns; and thirdly, experimental results demonstrating over 98% accuracy on CIFAR-100 classification, substantially outperforming existing quantum methods, which are typically limited to 70–80% accuracy.

Experimental validation shows robust performance across datasets: 99.2% validation accuracy on MNIST and 96.90% accuracy on CIFAR-100. The genetic algorithm optimization effectively identified optimal two-layer quantum circuit configurations with strategic entanglement patterns, demonstrating the potential for automated quantum architecture search.

To provide a realistic benchmark, we compared our model against the classical deep learning architecture ResNet-18, which achieved 99.2% accuracy on MNIST and 97.1% on CIFAR-100 in our experiments. These results confirm that the proposed HQCNN–REGA model is capable of approaching the performance of leading classical models.

This study addresses NISQ-era challenges by developing quantum algorithms that deliver practical benefits within current hardware constraints. The hybrid architecture provides a viable pathway for near-term quantum advantage in computer vision, with implications extending to other pattern recognition domains.

Several limitations suggest future research directions: extending to multi-class and larger datasets; optimizing genetic algorithm computational overhead; and testing on actual quantum hardware with noise and decoherence considerations. Future work should explore adaptive quantum–classical resource allocation and dynamic balancing strategies.

Our results demonstrate that carefully designed hybrid quantum–classical systems can achieve competitive performance on practical machine learning tasks, suggesting a promising trajectory toward quantum advantage in AI applications.

This research contributes to evidence that quantum machine learning can transition from theoretical promise to practical impact, particularly in high-dimensional data analysis and complex pattern recognition. The methodology presented lays a foundation for future quantum machine learning systems that harness quantum advantages while remaining compatible with modern classical ML workflows.

Author Contributions: Conceptualization, A.M. and B.D.; methodology, A.M.; software, A.M.; validation, A.M. and B.D.; formal analysis, A.M.; investigation, A.M.; resources, B.D.; data curation, A.M.; writing—original draft preparation, A.M.; writing—review and editing, A.M. and B.D.; visualization, A.M.; supervision, B.D.; project administration, B.D.; funding acquisition, B.D. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by Grant No. AP19576314 "Development of high-performance hybrid algorithms for solving oil displacement problems based on classical (GPU) and quantum computing" of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan.

Data Availability Statement: Dataset available upon request from the authors.

Conflicts of Interest: Authors Aksultan Mukhanbet and Beimbet Daribayev were employed by the company LLP "DigitAlem". The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
2. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
3. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Aaronson, S. Quantum Machine Learning Algorithms: Read the Fine Print. *Nat. Phys.* **2015**, *11*, 291–293. [CrossRef]
6. Preskill, J. Quantum Computing in the NISQ Era and Beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
7. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Lau, H.K.; Simonetto, A.; Kwek, L.C. Noisy Intermediate-Scale Quantum Algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
8. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef]
9. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum Principal Component Analysis. *Nat. Phys.* **2014**, *10*, 631–633. [CrossRef]
10. Mari, A.; Bromley, T.R.; Izaac, J.; Schuld, M.; Killoran, N. Transfer Learning in Hybrid Classical–Quantum Neural Networks. *Quantum* **2020**, *4*, 340. [CrossRef]
11. Grant, E.; Benedetti, M.; Cao, S.; Hallam, A.; Lockhart, J.; Stojevic, V.; Carleo, G.; Severini, S. Hierarchical Quantum Classifiers. *npj Quantum Inf.* **2018**, *4*, 65. [CrossRef]
12. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. *Nature* **2017**, *549*, 195–202. [CrossRef]
13. Dunjko, V.; Briegel, H.J. Machine Learning and Artificial Intelligence in the Quantum Domain. *Rep. Prog. Phys.* **2018**, *81*, 074001. [CrossRef]
14. Daribayev, B.; Mukhanbet, A.; Imankulov, T. Implementation of the HHL Algorithm for Solving the Poisson Equation on Quantum Simulators. *Appl. Sci.* **2023**, *13*, 11491. [CrossRef]
15. Schuld, M.; Fingerhuth, M.; Petruccione, F. Implementing a Distance-Based Classifier with a Quantum Interference Circuit. *EPL* **2017**, *119*, 60002. [CrossRef]
16. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028. [CrossRef]
17. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized Quantum Circuits as Machine Learning Models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [CrossRef]
18. Cong, I.; Choi, S.; Lukin, M.D. Quantum Convolutional Neural Networks. *Nat. Phys.* **2019**, *15*, 1273–1278. [CrossRef]
19. Henderson, M.; Shakya, S.; Pradhan, S.; Cook, T. Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. *Quantum Mach. Intell.* **2020**, *2*, 2. [CrossRef]
20. Liu, Y.; Arunachalam, S.; Gheorghiu, V.; Dunjko, V. Variational Quantum Circuits for Deep Reinforcement Learning. *IEEE Access* **2021**, *9*, 78269–78282.

21. Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J.I. Data Re-Uploading for a Universal Quantum Classifier. *Quantum* **2020**, *4*, 226. [CrossRef]
22. Griol-Barres, I.; Milla, S.; Cebrián, A.; Mansoori, Y.; Millet, J. Variational Quantum Circuits for Machine Learning. An Application for the Detection of Weak Signals. *Appl. Sci.* **2021**, *11*, 6427. [CrossRef]
23. Schuld, M.; Sweke, R. Effect of Data Encoding on the Expressive Power of Variational Quantum-Machine-Learning Models. *Phys. Rev. A* **2021**, *103*, 032430. [CrossRef]
24. Spector, L.; Barnum, H.; Bernstein, H.J.; Swamy, N. Finding a Better-than-Classical Quantum AND/OR Algorithm Using Genetic Programming. In Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 2239–2246.
25. Malossini, A.; Blanzieri, E.; Calarco, T. Quantum Genetic Optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 231–241. [CrossRef]
26. Cincio, L.; Rudinger, K.; Sarovar, M.; Coles, P.J. Machine Learning of Noise-Resilient Quantum Circuits. *PRX Quantum* **2021**, *2*, 010324. [CrossRef]
27. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren Plateaus in Quantum Neural Network Training Landscapes. *Nat. Commun.* **2018**, *9*, 4812. [CrossRef]
28. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum Circuit Learning. *Phys. Rev. A* **2018**, *98*, 032309. [CrossRef]
29. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational Quantum Algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
30. Huang, H.Y.; Kueng, R.; Preskill, J. Power of Data in Quantum Machine Learning. *Nat. Commun.* **2021**, *12*, 2631. [CrossRef] [PubMed]
31. Ramzan, F.; Khan, M.U.G.; Rehmat, A.; Iqbal, S.; Saba, T.; Rehman, A.; Mehmood, Z. A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. *J. Med. Syst.* **2020**, *44*, 37. [CrossRef]
32. Schuld, M. Supervised Quantum Machine Learning Models Are Kernel Methods. *arXiv* **2021**, arXiv:2101.11020. [CrossRef]
33. Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; Killoran, N. Quantum Embeddings for Machine Learning. *arXiv* **2020**, arXiv:2001.03622. [CrossRef]
34. Thumwanit, N.; Lortaraprasert, C.; Raymond, R. Invited: Trainable Discrete Feature Embeddings for Quantum Machine Learning. In Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 1352–1355.
35. Phalak, K.; Ghosh, A.; Ghosh, S. Optimizing Quantum Embedding Using Genetic Algorithm for QML Applications. *arXiv* **2024**, arXiv:2412.00286. [CrossRef]