Applied Soft Computing 141 (2023) 110307

Contents lists available at ScienceDirect

Applied Soft Computing

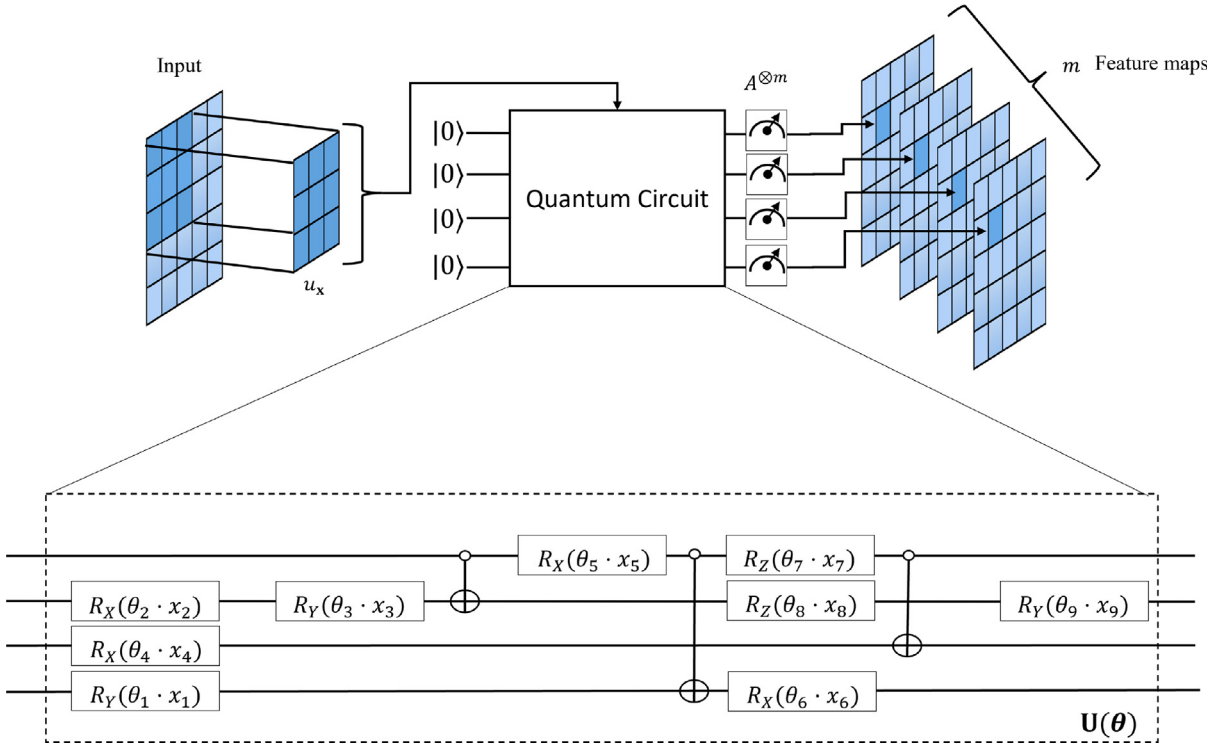
journal homepage: www.elsevier.com/locate/asoc

# Quantum angle encoding with learnable rotation applied to quantum–classical convolutional neural networks

Emmanuel Ovalle-Magallanes [a], Dora E. Alvarado-Carrillo [b], Juan Gabriel Avina-Cervantes [a,*], Ivan Cruz-Aceves [b], Jose Ruiz-Pinales [a]

[a] Telematics and Digital Signal Processing Research groups (CAs), Engineering Division, Campus Irapuato-Salamanca, University of Guanajuato, Carretera Salamanca-Valle de Santiago km 3.5 + 1.8 km, Comunidad de Palo Blanco, Salamanca 36885, Mexico
[b] Center for Research in Mathematics (CIMAT), A.C., Jalisco S/N, Col. Valenciana, Guanajuato 36000, Mexico

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Article history:*
Received 18 August 2022
Received in revised form 30 March 2023

## ABSTRACT

Quantum Machine Learning (QML) has experienced rapid progress in recent years due to the development of Noisy Intermediate-Scale Quantum (NISQ) devices and quantum simulators. Two key elements must be minimized To maintain acceptable computational complexity in QML: the number of qubits required to encode classical data and the number of quantum gates. This paper proposes a novel angle encoding with learnable rotation to drastically reduce the qubits and circuit depth from $\mathcal{O}(N)$

* Corresponding author.
*E-mail addresses:* e.ovallemagallanes@ugto.mx (E. Ovalle-Magallanes), dora.alvarado@cimat.mx (D.E. Alvarado-Carrillo), avina@ugto.mx (J.G. Avina-Cervantes), ivan.cruz@cimat.mx (I. Cruz-Aceves), pinales@ugto.mx (J. Ruiz-Pinales).

https://doi.org/10.1016/j.asoc.2023.110307
1568-4946/© 2023 Elsevier B.V. All rights reserved.

Accepted 6 April 2023
Available online 18 April 2023

*Keywords:*
Quantum angle encoding
Quantum–classical neural networks
Variational quantum circuits

to $\mathcal{O}(\lceil \log_2(N) \rceil)$ qubits, and only $N$ parameterized gates, where $N$ is the input size. Additionally, an extended quantum convolutional layer is introduced with multiple quantum circuits (quantum kernel) that allow for the configuration of any arbitrary size, stride, and dilation analogous to a classical convolutional layer. The proposed quantum convolutional layer learns multiple feature maps with a single quantum kernel while reducing computational cost by employing angle encoding with learnable rotation. Extensive experiments were performed by comparing diverse types of quantum convolutional configurations in a Quantum Convolutional Neural Network (QCNN) over a balanced subset of the MNIST and Fashion-MNIST datasets, achieving an accuracy of 0.90 and 0.7850, respectively.

© 2023 Elsevier B.V. All rights reserved.

## 1. Introduction

Convolutional Neural Networks (CNNs) have succeeded tremendously in various computer vision tasks such as image classification, object detection, image segmentation, and facial recognition [1–3]. To further improve these tasks, deep and handcrafted feature fusion has been proposed [4,5]. However, real-world applications are still challenging to solve with classical machine learning methods. In this sense, Quantum Machine Learning (QML) [6,7] has emerged as an interdisciplinary field that exploits the strengths of quantum computing to enhance classical machine learning algorithms.

In particular, Quantum Convolutional Neural Networks (QCNNs), also known as hybrid quantum–classical convolutional neural networks, are a family of QML that has recently become a very active field of research with the current availability of quantum devices in the noisy intermediate-scale quantum (NISQ) processors [8,9] and simulators [10]. The core idea of QCNNs is to apply classical CNN paradigms, including convolution, pooling, and linear layers, using quantum mechanical effects [11,12], such as superposition and entanglement, to improve the performance of the classical approach. Additionally, Quantum layers rely on Variational Quantum Circuits (VQC) [13], henceforth quantum kernels, to learn complex weights in high-dimensional Hilbert space.

Any quantum circuit requires three components: state preparation, unitary evolution, and measurement. First, the qubits of the quantum register are initialized. Then, an initial state is prepared by loading the classical data encoded in the register state. Afterward, parameterized and un-parameterized quantum gates perform unitary transformations (*i.e.,* quantum computations) on the register state. Finally, the quantum output state is measured and decoded from quantum to classical data. However, running a quantum circuit within many quantum gates that require multiple qubits remains impractical due to the restricted availability of qubits and vectorization techniques for quantum operations.

Therefore, research has focused on developing hybrid algorithms by including quantum-based routines in classical deep-learning models. This is the case with deep neural networks [14], non-trainable convolutional layers [15], trainable convolutional layers [16,17], and dense layers [18]. These quantum methods leverage a small number of qubits and shallow depth of quantum circuits to overcome the limitations of current quantum hardware and simulators. Although current quantum hybrid models have shown promising results, they face significant challenges when dealing with multi-quantum circuits of arbitrary size. One of the main constraints of quantum computing is that the size of quantum mappings and operations scales exponentially with the number of required qubits and quantum circuits, representing the main bottleneck in quantum computing. Hence, there is a need to develop new techniques that can improve their performance and scalability. Accordingly, this article proposes an extension to the capabilities of a quantum convolutional layer [15] by incorporating the ability to optimize multiple quantum kernels of arbitrary size, stride, and dilation, like classical convolutional layers.

Moreover, a novel quantum embedding technique is introduced, which reduces the required qubits substantially compared with the most used techniques, such as threshold [15] and angle embedding [19], from $\mathcal{O}(N)$ to $\mathcal{O}(\lceil \log_2(N) \rceil)$, where $N = k \times k$ and $k$ is the filter size. Ergo, the proposed technique allows training more general QCNNs. This contribution, in turn, improves performance for some tasks requiring more powerful processing (*i.e.,* a deeper model). Such a method is proven to guarantee both efficiency and effectiveness. Note that this quantum convolution maps a single input channel image into four channel feature maps with lower parameters than the previous version of quantum convolution while bringing performance gain. Empirical experiments were performed on the MNIST and Fashion-MNIST datasets to validate the proposed method, obtaining better accuracy and computational efficiency than existing QCNNs. Specifically, the main contributions of this study are as follows:

1. A novel quantum embedding based on simultaneous encoding and initial quantum state transformation is proposed to reduce computational complexity.
2. A lightweight and extended quantum convolutional layer is introduced to improve efficiency and enable greater configuration flexibility.
3. A sparse quantum circuit, built around the proposed embedding, is designed to alleviate the quantum convolutional layer bottleneck without compromising performance.

## 2. Related work

QML exploits unique quantum properties such as entanglement and superposition to improve components of classical machine learning, such as perceptrons and convolutions. The concept of quantum perceptron was introduced by Altaisky [20], who proposed to replace the input and hidden layers (including the activation function) with qubits and unitary operators, respectively. This analogy was demonstrated by Schuld et al. [21] and Cao et al. [22]. In these works, the quantum perceptron process multiple inputs in superposition, requiring additional register qubits; thus, the required qubits were doubled concerning the input data size. Following this idea, Wan et al. [23] proposed a quantum feed-forward network. First, the classical neurons were rendered by ancillary qubits. Then, they were generalized to being quantum reversible (*i.e.,* unitary). Finally, the quantum network was trained using gradient descent without any quantum state measurement. However, this network lacked non-linear activation functions, which were later added by Tacchino et al. [24] using non-linear measurements in each qubit, performed directly without needing an additional ancilla qubit. Zhao and Gao [14] introduced the Quantum Deep Neural Network (QDNN), which is analogous to a DNN. It consisted of three quantum layers: the input layer, the hidden layer, and the output layer, respectively. The input layer used an 8-qubit circuit, the hidden layer a 6-qubit circuit, and the output layer a 4-qubit circuit.

However, quantum devices (and simulators) have a limited number of qubits. As such, hybrid algorithms have been recently

**Table 1**
Summary of related work. Furthermore, giving input data of size $N$, the required embedding qubits are specified. n/a stands for not available.

| Author | Embedding technique | Required qubits | Dataset |
|---|---|---|---|
| Altaisky [20] | n/a | n/a | n/a |
| Schuld et al. [21] | Binary encoding | $N$ | n/a |
| Cao et al. [22] | Angle encoding | $N$ | n/a |
| Wan et al. [23] | n/a | n/a | n/a |
| Tacchino et al. [24] | Hypergraph States Generation Encoding | $\log_2(N)$ | Synthetic data |
| Zhao and Gao [14] | Angle encoding | $N$ | MNIST |
| Henderson et al. [15] | Thresholding Encoding | $N$ | MNIST |
| Mari et al. [18] | Angle encoding | $N$ | Hymenoptera |
| Liu et al. [16] | Angle encoding | $N$ | Tetris dataset |
| Chen [17] | Angle encoding | $N$ | MNIST and Fashion-MNIST |
| Ovalle-Magallanes et al. [25] | Angle encoding | $N$ | Coronary Stenosis patches |

proposed, holding both a quantum and a classical module. One approach is the quantum convolutional (or quanvolutional) layer introduced by Henderson et al. [15] to produce feature maps comparable to those generated by the classical convolutional counterpart by transforming classical data using Random Quantum Circuits (RQC). This quantum layer took as input spatially-local subsections of the image as a classical convolution operation. In this case, the weights of the quantum circuit remained idle. The layer was set at the bottom of the CNN (as the first layer) and decoded to feed classical layers. Afterward, Liu et al. [16] proposed a trainable version of the quanvolutional layer. This quantum layer employed VQCs instead, where the circuit parameters were optimized during the backward step. Later, Chen [17] incorporated the dilation property of the classical convolution layer into his quantum version. Thus, it allowed expanding the receptive field of the filters without increasing the kernel size. These last two methods employed a one-to-one variable qubit encoding and decoding scheme, generating multiple feature maps from a single quantum kernel.

Instead of plugging a quantum layer at the bottom of the network, Mari et al. [18] introduced a 4-qubit quantum layer placed at the top, before the SoftMax layer, to improve the final feature representation. This layer was jointly trained along the classical sub-module. Notice that this layer acts as a classical dense layer. In this direction, Ovalle-Magallanes et al. [25] proposed a quantum layer with a parallel 2- and 4-qubit VQC that reduced the computation time and processed a more significant number of classical features. This quantum layer was also placed at the top of the network (*i.e.,* after a global pooling layer and before the SoftMax layer).

Table 1 summarizes related works containing the encoding technique, the required number of qubits, and the employed dataset. It is worth mentioning that Table 1 does not compare the task performance, *i.e.,* image classification, as the quantum algorithms employed are diverse. Instead, it provides a general overview of the embedding strategies used. As discussed in previous paragraphs, hybrid quantum–classical algorithms have been developed to improve the performance of classical approaches. Nevertheless, they suffer from two main components that need to be minimal: the required qubits to encode the classical data and the number of quantum gates. Furthermore, one of the limitations of existing methods is that they can only use a single convolutional quantum kernel of a fixed size of $2 \times 2$. Thus, to the best of the authors' knowledge, using multiple kernels within the same convolutional quantum layer has not been successfully implemented in any publication.

This paper presents a novel encoding technique with learnable parameters that drastically reduces the quantum circuit computation complexity by simultaneously encoding the classical data and transforming the initial quantum state. Moreover, the proposed convolutional layer extends the current quantum convolutional layer's capabilities concerning tuning hyperparameters, such as kernel size, number of kernels, stride, and dilation.

## 3. Background

This section provides an overview of quantum computers' main components. First, the concepts and mathematical definitions of a qubit, quantum register, and quantum gates are described. Then, the core of quantum circuits, the fundamental building blocks of quantum algorithms, are presented.

### 3.1. Quantum primitives

The simplest unit of quantum computing is the quantum bit or qubit. The difference between bits and qubits is that a qubit can be in a superposition state such that:

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

where $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$, and $\{|0\rangle, |1\rangle\}$ are the qubit basis states (expressed in Dirac notation). These basis states are defined as:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2}$$

In general, the states of a quantum circuit of $n$-qubits are vectors in the $n$-qubit high-dimensional Hilbert space, such as:

$$|\psi\rangle = \sum_{i=0}^{2^{n-1}} \alpha_i |i\rangle, \quad s.t. \quad \sum_{i=0}^{2^{n-1}} |\alpha_i|^2 = 1, \tag{3}$$

where $|i\rangle$ is $|b_0 \cdots b_n\rangle$ with $b_0 \cdots b_n$ as the binary representation of $i$, and $\alpha_i$ is an amplitude of the quantum system.

A quantum computer employs quantum gates that act on either one or multiple qubits to manipulate or interact with such qubits. A quantum gate is represented by a unitary matrix $\mathbf{U}$, such that $\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = \mathbb{I}_n$, where $\mathbb{I}_n$ is the identity matrix in $\mathbb{R}^n$ and $\mathbf{U}^\dagger$ is the conjugate transpose. In this manner, a sequence of quantum gates forms a quantum circuit applied to an initial state. Table 2 illustrates the most frequently used quantum gates, circuit symbols, and mathematical expressions.

### 3.2. Quantum circuits

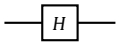
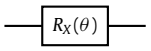
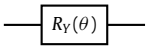
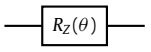
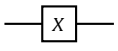
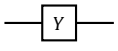
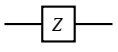
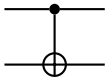
Quantum algorithms are typically hybrid and involve both classical and quantum computations. For instance, to map classical data onto a quantum circuit, a real vector $\mathbf{x}$ must be embedded in a quantum state $|\psi_\mathbf{x}\rangle$.

Let $E(\mathbf{x})$ be the encoding operator; the encoded quantum state is obtained by:

$$\mathcal{E} : \mathbf{x} \rightarrow |\psi_\mathbf{x}\rangle = E(\mathbf{x})|\psi_0\rangle, \tag{4}$$

where $|\psi_0\rangle$ is an initial state (*e.g.,* the ground state $|0\rangle^{\otimes n}$). The encoding determines how many qubits are required in the quantum circuit. Different encoding strategies have been proposed in prior

**Table 2**
Most employed quantum gates, showing their graphical and mathematical notation.

| Name | Circuit | Notation |
|---|---|---|
| Hadamard |  | $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ |
| X-Rotation |  | $R_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$ |
| Y-Rotation |  | $R_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$ |
| Z-Rotation |  | $R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$ |
| Pauli-X |  | $\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ |
| Pauli-Y |  | $\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$ |
| Pauli-Z |  | $\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ |
| CNOT |  | $CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ |
| Measurement |  | – |

literature, such as threshold encoding [15], angle encoding [19], and amplitude encoding [26].

Subsequently, the quantum gates perform unitary transformations on the encoded state, evolving the state to an output quantum state $|\psi_{\mathbf{y}}\rangle$ as follows:

$$\mathcal{U} : |\psi_{\mathbf{x}}\rangle \rightarrow |\psi_{\mathbf{y}}\rangle = \mathbf{U}(\boldsymbol{\theta})|\psi_{\mathbf{x}}\rangle, \tag{5}$$

where $\mathbf{U}$ is a sequence of executable unitary transformations of the form:

$$\mathbf{U}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \mathbf{V}_i(\boldsymbol{\theta}_i)\mathbf{W}_i, \tag{6}$$

where $\mathbf{W}_i$ are un-parameterized gates (*e.g.,* CNOT gates) and $\mathbf{V}_i(\boldsymbol{\theta}_i)$ are variational quantum circuits within a set of $q$ quantum gates (tensor products):

$$\mathbf{V}_i(\boldsymbol{\theta}_i) = \bigotimes_{j=1}^{q} R^{j,i}(\theta_{j,i}), \tag{7}$$

where $R^{j,i}$ is the $j$th rotation gate (one of the rotation gates $R_X, R_Y, R_Z$) acting on the $i$th qubit, and $\theta_{j,i}$ its respective rotation angle.

Finally, in this decoding stage, specific local observables $A^{\otimes m}$, such as the Pauli operator $\sigma_Z$, for $m \leq n$ qubits are measured in the output quantum state. Variance and expected value are examples of observables. Measurements can be made globally, where all qubits are measured, or locally, where only a few qubits or pairwise are measured. In such a way, the decoded data can be obtained through repeated measures of the chosen observable $A$ as:

$$\mathcal{M} : |\psi_{\mathbf{y}}\rangle \rightarrow \mathbf{y} = < \psi_{\mathbf{y}}|A^{\otimes m}|\psi_{\mathbf{y}} > . \tag{8}$$

Therefore, a particular quantum circuit design is a trade-off between two main components, which must be minimal: (1) the width or required qubits and (2) the number of quantum gates per unit transformation. Fig. 1 illustrates a typical structure of a quantum circuit.

## 4. Methods

Current quantum convolutional layers only support a single quantum kernel and are limited to a size of 2×2, with quadratic qubit requirements concerning kernel size; so, $n = 2 \times 2$ qubits are required. In contrast, this work presents an extended quantum convolutional layer where an arbitrary number of kernels, size, stride, and dilation can be configured. The proposed quantum convolutional layer grows logarithmically by the required qubits $n = \log_2(k \times k)$ and only needs $k \times k$ parameterized quantum gates, with $k$ as kernel size. Furthermore, an angle encoding with learnable rotation is introduced to achieve this saving. In this way, data encoding and quantum computation are performed simultaneously. Therefore, the number of quantum circuit executions during the quantum convolution process is drastically reduced.

### 4.1. Angle encoding with learnable rotation

Given its simplicity, angle encoding is the most widely used encoding approach, where single-qubit rotation gates encode classical input $\mathbf{x}$. Each element of the input determines the angle of the rotation gate (*e.g.,* an $R_Y$ rotation gate). As such, this approach requires $n$ qubits to encode $n$ input variables and can be defined as:

$$|\psi_{\mathbf{x}}\rangle = \bigotimes_{i=1}^{n} R(x_i)|\psi_0\rangle, \tag{9}$$

where $R$ is a rotation matrix.

$S$ sequential rotations (a unitary transformation) can be applied on the single-qubit $i$ to reduce the required qubits, as the form:

$$\mathbf{V}_i = \bigotimes_{j=1}^{S} R^{j,i}(x_j). \tag{10}$$

Notice that (7) and (10) share the same behavior, where a rotation gate receives an angle; however, in (10), the angle is fixed, *i.e.,* not trainable. By introducing a scale trainable factor to the rotation angle, (10) can be reformulated as:

$$\mathbf{V}_i(\boldsymbol{\theta}_i) = \bigotimes_{j=1}^{S} R^{j,i}(\theta_{j,i} \cdot x_j). \tag{11}$$

Therefore, the encoding is parameterized by $\boldsymbol{\theta}$, and the initial state $|\psi_0\rangle$ evolves through this parameterized rotation. Moreover, a dropout over the non-parameterized gates $\mathbf{W}$ is introduced during the unitary transformations to generate a sparse quantum circuit. Thus, (6) is re-written as:

$$\mathbf{U}(\boldsymbol{\theta}) = \prod_{i=1}^{n} \mathbf{V}_i(\boldsymbol{\theta}_i)(\delta_i \cdot \mathbf{W}_i), \tag{12}$$

where $\delta_i \sim \text{Bernoulli}(p)$ indicates that the random variable $\delta_i$ has the Bernoulli distribution with parameter $p$, where $0 < p < 1$. The proposed embedding employs a fixed $n = \lceil \log_2(k \times k) \rceil$ qubits and $(k \times k)$-rotation gates to map a classical input of size $\mathbf{x} \in \mathbb{R}^{k \times k}$. Consequently, rotation gates are re-arranged more deeply instead of wider (one-to-one).

The encoding step usually contains the Hadamard gate applied to the initial state, which leads to a uniform superposition state. Thereby, the proposed simultaneous encoding-evolution with learnable rotation is defined as:

$$\mathcal{U}^{\star} : |\psi_0\rangle \rightarrow |\psi_y\rangle = \mathbf{U}(\boldsymbol{\theta})(H|\psi_0\rangle), \tag{13}$$

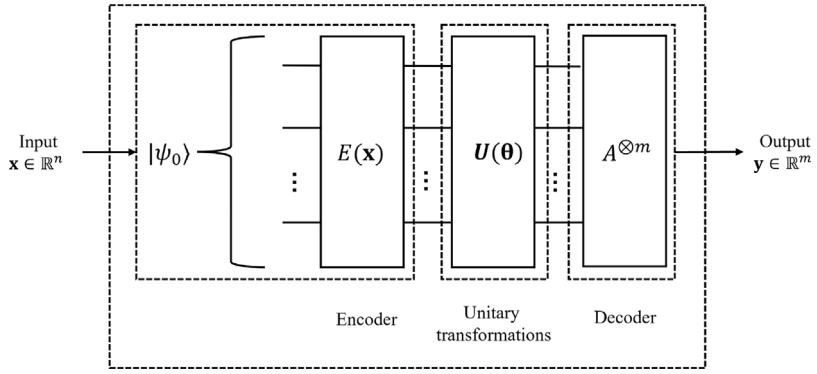



**Fig. 1.** Typical structure of a quantum circuit. An input data $\mathbf{x} \in \mathbb{R}^n$ is encoded and then transformed by a set of unitary transformations. Finally, $m$ measurements are carried out to decode the output data $\mathbf{y} \in \mathbb{R}^m$.

where $H$ stands for the Hadamard gate, and $\mathbf{U}(\boldsymbol{\theta})$ follows (12). Algorithm 1 summarizes this simultaneous encoding and transformation process, where the rotation gate is randomly selected, as well as the qubit where the operation is performed.

**Algorithm 1:** Circuits design of the Angle encoding with Learnable Rotation

**Data:** $\mathbf{x}$ ▷ Input data
**Result:** $|\psi_{\mathbf{y}}\rangle$ ▷ Output quantum state
$R \leftarrow [R_X, R_Y, R_Z]$ ▷ Set rotations gates
$\mathbf{W} \leftarrow$ CNOT ▷ Set non-parameterized gate
$H \leftarrow$ HADAMARD ▷ Set Hadamard gate
$p \leftarrow$ RAND(0, 1) ▷ Get dropout probability
$N \leftarrow$ SIZE($\mathbf{x}$) ▷ Get input size
$n \leftarrow \lceil \log_2(N) \rceil$ ▷ Get number of wires
$|\psi_0\rangle \leftarrow |0\rangle^{\otimes n}$ ▷ Set initial quantum state
$\boldsymbol{\theta} \leftarrow$ NORMAL(0, 1) $\in \mathbb{R}^N$ ▷ Set initial rotations
$\boldsymbol{\delta} \leftarrow$ BERNOULLI($p$) $\in \mathbb{R}^n$ ▷ Set initial Dropouts
$\mathbf{V} \leftarrow \emptyset$ ▷ Set unitary transformation list
$\mathbf{U} \leftarrow \emptyset$ ▷ Set quantum circuits list
**for** $j \leftarrow 0$ **to** $N$ **do**
  rnd_idx = RAND(0, 2) ▷ Random rotation index
  rnd_wire = RAND(0, $n$) ▷ Random wire index
  $R_\sigma = R$[rnd_idx] ▷ Select random rotation gate
  $\mathbf{V}$[rnd_wire] $= R_\sigma(\boldsymbol{\theta}_j \cdot x_j)$ ▷ Append operation
**for** $i \leftarrow 0$ **to** $n$ **do**
  $\mathbf{U}[i] = \mathbf{V}(\boldsymbol{\theta}_i)(\boldsymbol{\delta}_i \cdot \mathbf{W}_i)$ ▷ Append operation
$|\psi_{\mathbf{y}}\rangle = \mathbf{U}(\boldsymbol{\theta})(H|\psi_0\rangle)$ ▷ Compute output quantum state

### 4.2. Extended quantum convolution layer

One of the critical components of CNNs is the convolutional layer, which applies a 2D convolution operation between an input image $\mathbf{x} \in \mathbb{R}^{H_{in} \times W_{in}}$ and a kernel $\mathbf{w} \in \mathbb{R}^{k_h \times h_w}$ of learnable weights. In general, square and odd kernels are used, such that $k = k_h = k_w$.

Let $\mathbf{y} \in \mathbb{R}^{H_{out} \times W_{out}}$ be the result of a convolutional layer known as a feature map. Each element $y(i, j)$ is given by:

$$y(i,j) = \mathbf{w} * x(i + r, j + r), \tag{14}$$

$$y(i,j) = \sum_{u=1}^{k_h} \sum_{v=1}^{k_w} x(i + r\,u, j + r\,v)\, w(u, v), \tag{15}$$

where $*$ is the convolution operator and $r$ is the dilation rate. If $r = 1$, a dilated convolution becomes a standard convolution. A dilated convolution enlarges the receptive field without increasing the number of parameters or the amount of computation. For example, a small size kernel with a filter $k \times k$ expands to $k + (k - 1)(r - 1)$ with step length $r > 1$.

The spatial resolution of the resulting feature map can be calculated as follows:

$$H_{out} = \left\lfloor \frac{H_{in} + 2\,p - r\,(k_h - 1) - 1}{s} + 1 \right\rfloor, \tag{16}$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2\,p - r\,(k_w - 1) - 1}{s} + 1 \right\rfloor, \tag{17}$$

where $p$ is the padding determining how many zero values are added to the image border, and $s$ is the stride with which the input pixels are sampled. Thus, a set of $K$ kernels produces a feature map of $K$ channels. Moreover, activation functions are applied after the convolution operation to incorporate non-linearity into the network.

Let $u_{\mathbf{x}}$ be a local region of size $k \times k$ in the input where a particular kernel applies a convolution operation with a given stride, dilation, and size. Herein, the classical convolutional kernel is replaced by a variational quantum circuit that takes as input the local region $u_{\mathbf{x}}$ and processes it using a series of parameterized quantum gates to generate the output, as shown in Fig. 2. In this way, $u_{\mathbf{x}}$ feeds a quantum convolution of $n$-qubits, where a single quantum circuit can recover up to $m \leq n$ feature maps; that is, $\frac{K}{m}$ quantum circuits are required to obtain $K$ feature maps as a classical convolutional layer. Note that the number of gates must be set to $N = k \times k$ to have the same parameters as a classical kernel of size $k \times k$.

Since the input and output of a quantum circuit are classical values, all the encoding, transformation, and decoding of the quantum convolutional can be defined as:

$$\mathcal{Q}(\mathbf{x}; \boldsymbol{\theta}) : \mathbf{x} \in \mathbb{R}^N \rightarrow \mathbf{y} \in \mathbb{R}^m = \mathcal{M} \circ \mathcal{U}^\star \circ \mathcal{E}, \tag{18}$$

where the parameters $\boldsymbol{\theta}$ can be updated using optimization algorithms. The process is repeated for each local region of the input data, producing a set of feature maps that can be used for subsequent processing steps.

In a deep learning context, $\mathcal{Q}(\mathbf{x}; \boldsymbol{\theta})$ can be seen as a layer in a deep neural network [15]. Furthermore, it can be embedded in a classical CNN as follows:

$$\mathcal{N} = \mathcal{L}(\mathbf{x}^{(l)}, \boldsymbol{\theta}^{(l)}) \circ \mathcal{L}(\mathbf{x}^{(l-1)}, \boldsymbol{\theta}^{(l-1)}) \circ \cdots \circ \mathcal{L}(\mathbf{x}^{(1)}, \boldsymbol{\theta}^{(1)}), \tag{19}$$

where each layer $\mathcal{L}(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)})$ is a classical or a quantum layer.

## 5. Results and discussion

### 5.1. Datasets

In this work, two image datasets were used: MNIST and Fashion-MNIST. The MNIST database (Modified National Institute

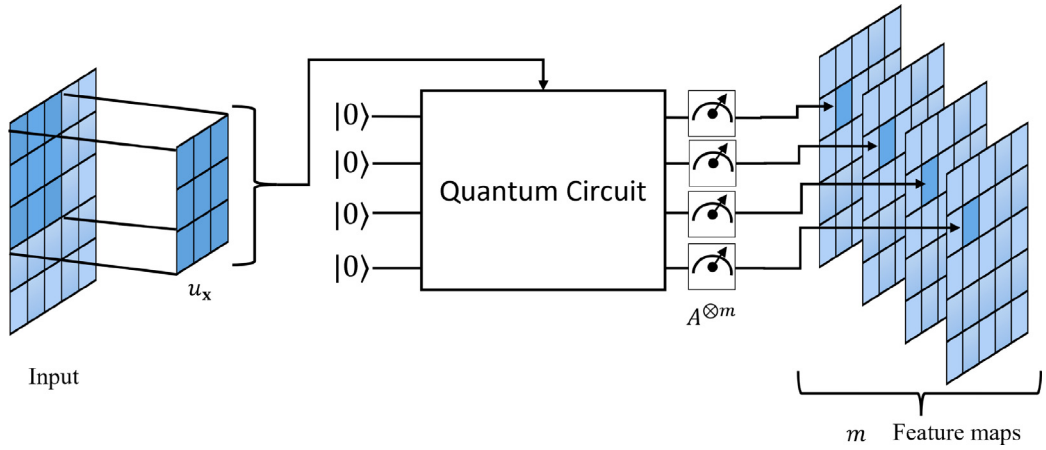



**Fig. 2.** Quantum convolution operation. A quantum circuit acts like a classical kernel over an image region $u_{\mathbf{x}}$, retrieving multiple feature maps from a single quantum circuit.

of Standards and Technology database) [27] is a collection of handwritten digits from '0' to '9', which contains 70,000 images. Each image is grayscale and 28 × 28 pixels in size. It has a training set of 60,000 examples and a test set of 10,000 examples. The Fashion-MNIST [28] has fashion image products from 10 categories, with 7,000 images per category. Fashion-MNIST shares the exact image size, data format, and the structure of training and testing split with the original MNIST.

Due to the extensive computing requirements of the quantum elements (*e.g.,* limited availability of qubits and un-vectorization techniques for quantum operations) from both datasets, a balanced fraction was selected for training and validation. As a result, the training dataset consists of 1200 and 200 samples for testing.

### 5.2. Implementation details

The employed architectures of the QCNN consist of one quantum convolutional layer, with a stride of two, $q$-quantum kernels $q \in (1, 2, 3)$, a kernel size $k \in (3, 5)$, and a dilation ratio $r \in (1, 2, 3)$. Fig. 3 shows the schematic image of the 3 × 3 quantum kernels configurations. Similarly, Fig. 4 shows the corresponding schematic diagrams of the 5 × 5 quantum kernels configurations. Notice that different numbers of each rotation gate can be selected, and the quantum circuit maybe not contain CNOT gates due to the stochastic nature of the circuit design. Consequently, it extracts from the 28 × 28 input image a feature map $\mathbf{F} \in \mathbb{R}^{14\times14\times nq}$, where $n = \lceil \log_2(k \times k) \rceil$. The padding was set accordingly to fit this output size. Then, $\mathbf{F}$ is flattened to feed a fully connected layer with ten outputs to obtain the probabilities of each class through a SoftMax activation.

Moreover, the quantum layer was configured with an initial quantum state $|0\rangle^{\otimes n}$, a dropout probability for non-parametric gates of $p = 0.3$, and $k \times k$ parameterized rotation gates. Finally, in the decoding module, a Pauli Z operator is applied over each wire of the quantum circuit. That is, for each qubit, an output is generated.

The model was trained with a batch size of 32 for 20 epochs employing the Adam optimizer with a learning rate of $1 \times 10^{-2}$ that minimizes the Cross-Entropy Loss. All experiments were performed using Pytorch and the PennyLane Quantum library [10] in a 24 GB NVIDIA Titan RTX GPU.

### 5.3. Ablation study

An ablation study evaluated the impact of the proposed quantum angle encoding applied within a quantum convolutional layer. Table 3 shows the test accuracies and losses achieved by the different QCNNs configurations among the two studied datasets. Additionally, the average training time per epoch is detailed. Table 3 shows that for MNIST dataset, the best quantum convolutional layer with a kernel size of 3 × 3 achieved an accuracy of 0.90 and a loss of 0.34. The layer includes two kernels and a dilation ratio of 3. This strategy represented a boost of 7.5% in accuracy when only a single kernel with no dilation was employed. Also, for the case of a kernel size of 5 × 5, two

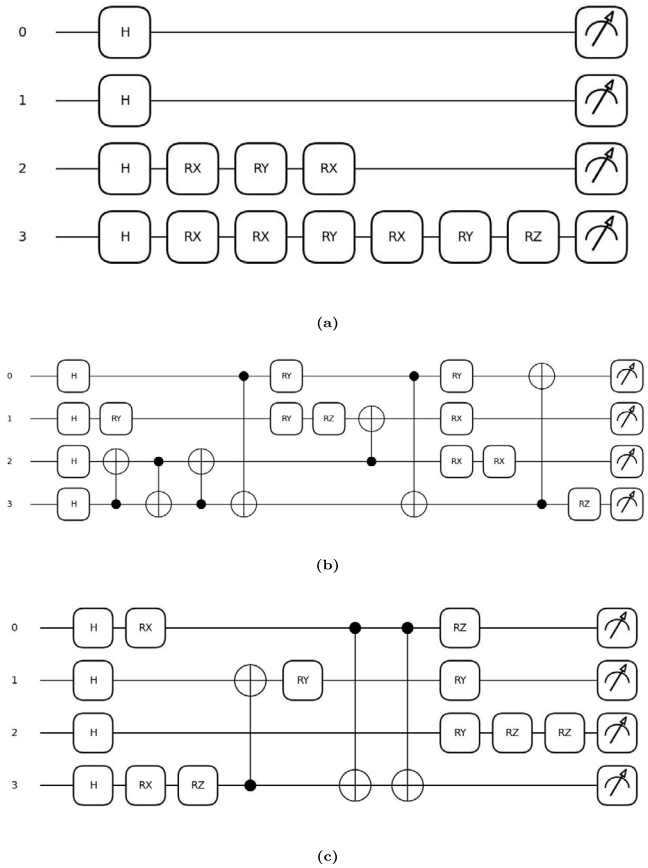



**Fig. 3.** 3 × 3 quantum circuits employed in the convolutional quantum layer. For a single kernel configuration, circuit (a) is applied for two kernels, circuits (a) and (b), and for three kernels, circuits (a), (b), and (c).

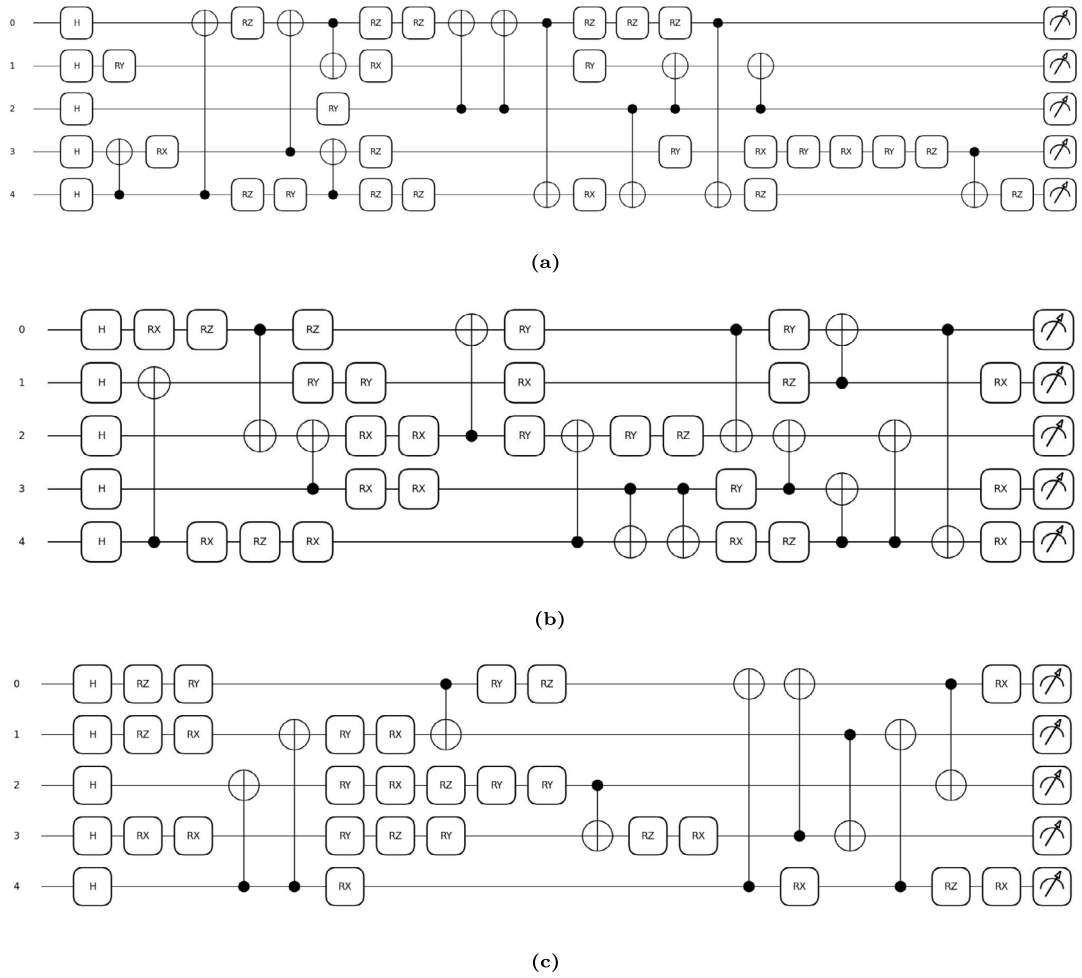


**Fig. 4.** $5 \times 5$ quantum circuits employed in the convolutional quantum layer. For a single kernel configuration, circuit (a) is applied for two kernels, circuit (a) and (b), and for three kernels, circuit (a), (b), and (c).

kernels and a dilation ratio of 2 obtained an accuracy gain of 23% (reaching 0.85%) concerning the single kernel and no dilation.

In the Fashion-MNIST dataset, if a kernel size of $3 \times 3$, an accuracy up to 0.78 was achieved using a single kernel with a dilation ratio of 3, with a 6.5% gain concerning the basic configuration (one kernel and non-dilation). On the other hand, if the kernel size is $5 \times 5$, an accuracy up to 0.77 was achieved by employing three quantum kernels and a dilation ratio of $r = 1$. This configuration outperforms the single quantum kernel layer results by 3%.

### 5.4. Classification comparison

The best configurations from the ablation study were compared with another QCNN technique from the state-of-the-art. Numerical results are shown in Table 4, where for the MNIST dataset, the proposed model achieved the best test accuracy and loss. In the case of Fashion-MNIST, the proposed model obtained competitive results in test accuracy and the second-best test loss. Regarding training time, the size and number of quantum kernels contribute directly. Thus, smaller and fewer kernels attained lower training times.

Nevertheless, the proposed encoding technique was able to reduce the required qubits per kernel from $\mathcal{O}(n)$ to $\mathcal{O}(\log_2(n))$. Besides, a single quantum rotation gate is required by variable to reduce the computational complexity concerning the number of quantum gates, and a dropout ratio is employed to include or not CNOT gates.

Moreover, including this encoding into a quantum convolutional layer enhances the current capabilities. For example, the introduced quantum convolutional layer can create quantum kernels of any size, stride, and dilation like a classical convolutional layer. Additionally, in the quantum convolution of $n$-qubits, each single quantum circuit can recover up to $m \leq n$ feature maps; that is, $\frac{K}{m}$ quantum circuits are required to obtain $K$ feature maps as a classical convolutional layer. Furthermore, the proposed quantum layer is configured straightforwardly to be included in any position of a CNN. However, considering the required training time per kernel, which still manifests some drawbacks due to the non-vectorized techniques available for quantum computation, it is common to find this layer as the first layer of a model.

In particular, two main limitations need to be addressed; firstly, although the number of required qubits is $\log_2(N)$, the depth of the quantum circuits grows exponentially; thus, it is necessary to create balanced circuits to reduce the depth, *i.e.,* each qubit has the same number of gates to perform the computations [29]. Secondly, the proposed model is currently conducted in the quantum simulator, and now that small qubit-real quantum computers are available (IBM Quantum [30] and Rigetti [31]), a performance comparison could be resolved in future research.

## 6. Conclusion and future work

This work proposed a novel quantum angle embedding with learnable rotation, drastically reducing the required qubits of a

**Table 3**
Ablation study results employing different quantum convolutional layer configurations.

| Dataset | Kernel size | Num. kernels | Dilation ratio | Test acc. | Test loss | Epoch time [s] |
|---|---|---|---|---|---|---|
| MNIST | 3 × 3 | 1 | 1 | 0.8250 | 0.6087 | ≈ 2080 |
| | | | 2 | 0.8150 | 0.5484 | ≈ 2112 |
| | | | 3 | 0.7950 | 0.6115 | ≈ 2112 |
| | | 2 | 1 | 0.8750 | 0.4223 | ≈ 4192 |
| | | | 2 | **0.9000** | **0.3441** | ≈ 4256 |
| | | | 3 | 0.8750 | 0.4179 | ≈ 4224 |
| | | 3 | 1 | 0.8450 | 0.4933 | ≈ 6336 |
| | | | 2 | 0.8550 | 0.3738 | ≈ 6368 |
| | | | 3 | 0.8550 | 0.4676 | ≈ 6368 |
| | 5 × 5 | 1 | 1 | 0.6200 | 1.0923 | ≈ 9568 |
| | | | 2 | 0.7500 | 0.6956 | ≈ 9632 |
| | | | 3 | 0.8200 | 0.5459 | ≈ 9664 |
| | | 2 | 1 | 0.7200 | 0.8659 | ≈ 19232 |
| | | | 2 | 0.8500 | 0.5021 | ≈ 19328 |
| | | | 3 | 0.8350 | 0.5238 | ≈ 19424 |
| | | 3 | 1 | 0.7400 | 0.7729 | ≈ 28544 |
| | | | 2 | 0.7950 | 0.6178 | ≈ 28992 |
| | | | 3 | 0.8500 | 0.5155 | ≈ 28896 |
| Fashion-MNIST | 3 × 3 | 1 | 1 | 0.7200 | 0.8331 | ≈ 1984 |
| | | | 2 | 0.7400 | 0.7662 | ≈ 1856 |
| | | | 3 | **0.7850** | 0.7306 | ≈ 1824 |
| | | 2 | 1 | 0.7650 | 0.7072 | ≈ 3872 |
| | | | 2 | 0.7600 | 0.7059 | ≈ 3680 |
| | | | 3 | 0.7700 | 0.7418 | ≈ 3680 |
| | | 3 | 1 | 0.7500 | 0.7138 | ≈ 5472 |
| | | | 2 | 0.7600 | **0.6959** | ≈ 5536 |
| | | | 3 | 0.7450 | 0.7526 | ≈ 5472 |
| | 5 × 5 | 1 | 1 | 0.7400 | 0.7636 | ≈ 9696 |
| | | | 2 | 0.7100 | 0.8158 | ≈ 9696 |
| | | | 3 | 0.7500 | 0.7833 | ≈ 9696 |
| | | 2 | 1 | 0.7350 | 0.8099 | ≈ 19328 |
| | | | 2 | 0.7250 | 0.7988 | ≈ 19168 |
| | | | 3 | 0.7650 | 0.7331 | ≈ 19136 |
| | | 3 | 1 | 0.7700 | 0.7866 | ≈ 29440 |
| | | | 2 | 0.7650 | 0.7513 | ≈ 28608 |
| | | | 3 | 0.7600 | 0.7285 | ≈ 28704 |

**Table 4**
Classification comparison of QCNN on the MNIST and Fashion-MNIST.

| Dataset | Model | Test acc. | Test loss | Epoch time [s] |
|---|---|---|---|---|
| MNIST | Henderson et al. [15] | 0.8800 | 0.4389 | – |
| | Liu et al. [16] | 0.8650 | 0.5341 | 1386 |
| | Chen [17] | 0.8950 | 0.3646 | **1095** |
| | Proposed | **0.9000** | **0.3441** | 4256 |
| Fashion-MNIST | Henderson et al. [15] | 0.7850 | **0.6919** | – |
| | Liu et al. [16] | 0.7950 | 0.8923 | 1392 |
| | Chen [17] | **0.8050** | 0.8837 | **1191** |
| | Proposed | 0.7850 | 0.7306 | 1824 |

quantum circuit. An extended quantum convolutional layer was introduced, mimicking the behavior of a classical convolutional layer. It receives any stride, dilation ratio, kernel size, and the number of kernels. In particular, the quantum convolutional layer employed the proposed embedding to require fewer qubits and quantum circuits. In this sense, the quantum circuit design includes a dropout element to control the non-parameterized gates (CNOT); thus, sparse circuits were generated. It is worth noting that the encoding and unitary transformations co-occur, reducing the required operations. Also, notice that a single quantum kernel computes multiple feature maps compared to classical kernels operating in a one-to-one relation. Numerical results demonstrate the potential of the proposed embedding and quantum layer, showing high accuracy rates on both MNIST and Fashion-MNIST datasets.

Further research is required on whether a more suitable QCNN can be constructed to further increase the performance, *e.g.*, through a different circuit design, considering the trade-off between the required qubits and the circuit depth. Also, in future research, different quantum measurement strategies should be studied.

## CRediT authorship contribution statement

**Emmanuel Ovalle-Magallanes:** Conceptualization, Methodology, Software, Writing – original draft. **Dora E. Alvarado-Carrillo:** Data curation, Methodology, Writing – original draft. **Juan Gabriel Avina-Cervantes:** Writing – review & editing, Investigation, Data curation. **Ivan Cruz-Aceves:** Visualization, Supervision, Investigation. **Jose Ruiz-Pinales:** Software, Data curation, Validation.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Juan Gabriel Avina-Cervantes reports financial support, administrative support, equipment, drugs, or supplies, and writing assistance were provided by University of Guanajuato. Emmanuel Ovalle- Magallanes reports financial support, administrative support, and equipment, drugs, or supplies were provided by National Council on Science and Technology (CONACyT). Juan Gabriel Avina-Cervantes reports a relationship with University of Guanajuato that includes: employment and funding grants. Emmanuel Ovalle-Magallanes reports a relationship with National Council on Science and Technology (CONACyT) that includes: funding grants.

## Data availability

Data will be made available on request.

## Acknowledgments

This work was supported in part by the University of Guanajuato CIIC (Convocatoria Institucional de Investigación Científica, UG) Project 094/2023 and Grant NUA 147347. Partially by the Mexican Council of Science and Technology CONACyT (Grants no. 626154/755609 and 626155/719327), the Bajio Supercomputer Laboratory (Grant no. 300832), and by the Mexican National Council of Science and Technology under project Cátedras-CONACyT No. 3150-3097.

## References

[1] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, Y. Miao, Review of image classification algorithms based on convolutional neural networks, Remote Sens. 13 (22) (2021) 4712, http://dx.doi.org/10.3390/rs13224712.
[2] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: Analysis, applications, and prospects, IEEE Trans. Neural Netw. Learn. Syst. (2021) http://dx.doi.org/10.1109/TNNLS.2021.3084827.
[3] G.W. Lindsay, Convolutional neural networks as a model of the visual system: Past, present, and future, J. Cogn. Neurosci. 33 (10) (2021) 2017–2031, http://dx.doi.org/10.1162/jocn_a_01544.
[4] B. Jena, S. Saxena, G.K. Nayak, L. Saba, N. Sharma, J.S. Suri, Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review, Comput. Biol. Med. 137 (2021) 104803, http://dx.doi.org/10.1016/j.compbiomed.2021.104803.
[5] S. Walia, K. Kumar, M. Kumar, X.-Z. Gao, Fusion of handcrafted and deep features for forgery detection in digital images, IEEE Access 9 (2021) 99742–99755, http://dx.doi.org/10.1109/ACCESS.2021.3096240.
[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Quantum machine learning, Nature 549 (7671) (2017) 195–202, http://dx.doi.org/10.1038/nature23474.
[7] M. Schuld, Supervised Learning with Quantum Computers, first ed., Springer, New York, NY, 2018, http://dx.doi.org/10.1007/978-3-319-96424-9.

[8] S. Bravyi, D. Gosset, R. Koenig, M. Tomamichel, Quantum advantage with noisy shallow circuits, Nat. Phys. 16 (10) (2020) 1040–1045, http://dx.doi.org/10.1038/s41567-020-0948-z.
[9] S. Wei, Y. Chen, Z. Zhou, G. Long, A quantum convolutional neural network on NISQ devices, AAPPS Bulletin 32 (1) (2022) 1–11, http://dx.doi.org/10.1007/s43673-021-00030-3.
[10] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M.S. Alam, S. Ahmed, J.M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, et al., Pennylane: Automatic differentiation of hybrid quantum-classical computations, 2018, arXiv preprint arXiv:1811.04968, https://arxiv.org/abs/1811.04968.
[11] M.A. Nielsen, I.L. Chuang, Quantum computation and quantum information, Amer. J. Phys. 70 (5) (2002) 558, http://dx.doi.org/10.1119/1.1463744.
[12] R. De Wolf, Quantum computing: Lecture notes, 2019, arXiv preprint arXiv:1907.09415, https://arxiv.org/abs/1907.09415.
[13] M. Benedetti, E. Lloyd, S. Sack, M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. 4 (4) (2019) 043001, http://dx.doi.org/10.1088/2058-9565/ab4eb5.
[14] C. Zhao, X.-S. Gao, QDNN: deep neural networks with quantum layers, Quantum Mach. Intell. 3 (1) (2021) 1–9, http://dx.doi.org/10.1007/s42484-021-00046-w.
[15] M. Henderson, S. Shakya, S. Pradhan, T. Cook, Quanvolutional neural networks: powering image recognition with quantum circuits, Quantum Mach. Intell. 2 (1) (2020) 1–9, http://dx.doi.org/10.1007/s42484-020-00012-y.
[16] 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) 1–8, http://dx.doi.org/10.1007/s11433-021-1734-3.
[17] Y. Chen, Quantum dilated convolutional neural networks, IEEE Access 10 (2022) 20240–20246, http://dx.doi.org/10.1109/ACCESS.2022.3152213.
[18] A. Mari, T.R. Bromley, J. Izaac, M. Schuld, N. Killoran, Transfer learning in hybrid classical-quantum neural networks, Quantum 4 (2020) 340, http://dx.doi.org/10.22331/q-2020-10-09-340.
[19] E.M. Stoudenmire, D.J. Schwab, Supervised learning with tensor networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS '16, Curran Associates Inc., Red Hook, NY, USA, 2016, pp. 4806–4814, http://dx.doi.org/10.5555/3157382.3157634.
[20] M. Altaisky, Quantum neural network, 2001, ArXiv Preprint Quant-Ph/0107012, https://arxiv.org/pdf/quant-ph/0107012.
[21] M. Schuld, I. Sinayskiy, F. Petruccione, Simulating a perceptron on a quantum computer, Phys. Lett. A 379 (7) (2015) 660–663, http://dx.doi.org/10.1016/j.physleta.2014.11.061.
[22] Y. Cao, G.G. Guerreschi, A. Aspuru-Guzik, Quantum neuron: an elementary building block for machine learning on quantum computers, 2017, arXiv preprint arXiv:1711.11240, https://arxiv.org/pdf/1711.11240.
[23] K.H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, M. Kim, Quantum generalisation of feedforward neural networks, Npj Quantum Inf. 3 (1) (2017) 1–8, http://dx.doi.org/10.1038/s41534-017-0032-4.
[24] F. Tacchino, P. Barkoutsos, C. Macchiavello, I. Tavernelli, D. Gerace, D. Bajoni, Quantum implementation of an artificial feed-forward neural network, Quantum Sci. Technol. 5 (4) (2020) 044010, http://dx.doi.org/10.1088/2058-9565/abb8e4.
[25] E. Ovalle-Magallanes, J.G. Avina-Cervantes, I. Cruz-Aceves, J. Ruiz-Pinales, Hybrid classical-quantum convolutional neural network for stenosis detection in X-ray coronary angiography, Expert Syst. Appl. 189 (2022) 116112, http://dx.doi.org/10.1016/j.eswa.2021.116112.
[26] R. LaRose, B. Coyle, Robust data encodings for quantum classifiers, Phys. Rev. A 102 (3) (2020) 032420, http://dx.doi.org/10.1103/PhysRevA.102.032420.
[27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.
[28] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017, arXiv preprint arXiv:1708.07747, https://arxiv.org/abs/1708.07747.
[29] I.F. Araujo, D.K. Park, F. Petruccione, A.J. da Silva, A divide-and-conquer algorithm for quantum state preparation, Sci. Rep. 11 (1) (2021) 1–12, http://dx.doi.org/10.1038/s41598-021-85474-1.
[30] IBM Quantum, IBM quantum, 2022, Retrieved from https://quantum-computing.ibm.com/. (Accessed 29 March).
[31] Rigetti, Think quantum, 2022, Retrieved from https://www.rigetti.com/. (Accessed 29 March).