Hindawi
Quantum Engineering
Volume 2022, Article ID 5701479, 9 pages
https://doi.org/10.1155/2022/5701479

Research Article

An Image Classification Algorithm Based on Hybrid Quantum Classical Convolutional Neural Network

Wei Li,[1] Peng-Cheng Chu,[2] Guang-Zhe Liu,[2] Yan-Bing Tian,[1] Tian-Hui Qiu ,[2] and Shu-Mei Wang [2]

[1] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao, China*
[2] *School of Science, Qingdao University of Technology, Qingdao, China*

Correspondence should be addressed to Tian-Hui Qiu; qiutianhui@qut.edu.cn and Shu-Mei Wang; wangshumei@qut.edu.cn

Received 21 April 2022; Revised 30 May 2022; Accepted 21 June 2022; Published 14 July 2022

Academic Editor: ShiJie Wei

Quantum machine learning is emerging as a strategy to solve real-world problems. As a quantum computing model, parameterized quantum circuits provide an approach for constructing quantum machine learning algorithms, which may either realize computational acceleration or achieve better algorithm performance than classical algorithms. Based on the parameterized quantum circuit, we propose a hybrid quantum-classical convolutional neural network (HQCCNN) model for image classification that comprises both quantum and classical components. The quantum convolutional layer is designed using a parameterized quantum circuit. It is used to perform linear unitary transformation on the quantum state to extract hidden information. In addition, the quantum pooling unit is used to perform pooling operations. After the evolution of the quantum system, we measure the quantum state and input the measurement results into a classical fully connected layer for further processing. We demonstrate its potential by applying HQCCNN to the MNIST dataset. Compared to a convolutional neural network in a similar architecture, the results reveal that HQCCNN has a faster training speed and higher testing set accuracy than a convolutional neural network.

1. Introduction

Quantum mechanics is a set of rules constructed by mathematical framework or physical theory. As a new computational framework for integrating the concepts of quantum mechanics, quantum computing [1,2] utilizes the properties of quantum mechanics to process both quantum and classical information [3,4], which may cause an insurmountable gap between quantum and classical computers. As quantum technology is developing, noisy intermediate-scale quantum computers (NISQ) have begun to deal with some relatively complex computing tasks, and their computing power has surpassed classical computers in some tasks [5]. At this stage, quantum algorithms are dedicated to implementation on NISQ devices, which gives a new direction to some real classical computational problems.

In parallel to the recent advances in classical machine learning, interest in quantum machine learning has grown significantly in academia [6–9]. As a quantum computing model, parameterized quantum circuits (PQCs) have powerful parallel computing and expressive capabilities. Many researchers have constructed quantum machine learning algorithms using PQC. In 2014, Peruzzo et al. [10] proposed a variational quantum eigensolver (VQE) and realized it with a photon quantum processor and a classical computer. In 2016, McClean et al. [11] improved the variational quantum algorithm's optimization scheme. In 2020, Schuld et al. [12] used PQC to construct a circuit-centric quantum classifier. The simulation on the dataset showed that the quantum classifier had high performance. In 2022, Zeng et al. [13] proposed a hybrid quantum neural network model based on a ladder circuit structure and introduced a full measurement strategy. The model achieved stable and high accuracy in both binary and multiclassification tasks. Since PQC was first proposed, many quantum machine learning algorithms based on PQC have been proposed [14–20], and it has been proved that the quantum machine learning algorithm based on PQC has better expressive performance and computing ability than the classical algorithm [21,22].

Convolutional neural networks (CNNs) have demonstrated their strong performance in deep learning tasks. Weight sharing and local receptive fields enable CNN to perform most tasks nowadays [23–28]. Furthermore, with the development of quantum information, quantum convolutional neural networks (QCNNs) have begun to interest researchers [29,30]. In 2019, inspired by classical CNN, Cong et al. [31] proposed a QCNN for phase classification and optimization of quantum error correction codes. The QCNN had a similar network architecture to CNN. In 2020, Li et al. [32] proposed a quantum deep convolutional neural network (QDCNN), which was a quantum version of classical convolution and achieved exponential computational acceleration. The model demonstrated the advantages of a quantum system over a classical system. In 2021, Parthasarathy et al. [33] proposed a quantum optical convolutional neural network that integrated quantum convolution and quantum optics. First, salient information was extracted using the linear unitary transformation of the quantum convolution part; subsequently, the quantum convolution results were input into the quantum optical neural network for further processing. The network exhibited good stability in terms of image recognition.

Combining PQC with the classical neural network, we propose a hybrid quantum-classical model for image classification. The hybrid model not only utilizes the powerful performance of PQC but also inherits the characteristics of the classical neural network. The quantum convolutional layer is composed of quantum convolution kernels constructed by PQC. We transform the classic image into a quantum state and input it into the quantum convolutional layer. The quantum convolutional layer performs the unitary transformation on the qubits corresponding to the convolution window. Then, the quantum pooling layer composed of quantum pooling units performs pooling operations. The qubits are measured at specific locations to obtain the results of the evolution of the quantum system, and the results are input to a fully connected layer for further processing. In this study, we refer to this model as a hybrid quantum-classical convolutional neural network (HQCCNN); the simulation experiments show that HQCCNN exhibits strong learning ability and high image classification accuracy.

The organization of the paper is as follows: CNN and PQC models are introduced in Section 2 briefly. The HQCCNN architectures and the algorithm execution process are described in Section 3. Section 4 describes simulation and analysis. Finally, the conclusion is drawn in Section 5.

2. Background

2.1. Convolutional Neural Network. A CNN consists of convolutional layers, pooling layers, and a fully connected layer that outputs the image's class. The convolution process is to convolve the convolution kernel with the image and then move the convolution kernel according to the step size (usually set to 1); the convolution kernel does the same operation.

The convolution operation can be defined as follows: suppose the input image A is the size of $m \times n$, the convolution kernel P is the size of $w \times w$, and the convolution operation can be completed through the equation

$$f = \sigma\left(A_{i:i+w,j:\ j+w}P\right). \tag{1}$$

Here, σ is the nonlinear function. In order to set the feature map's size to be the same as the input image, padding is usually used in the convolutional layer.

After convolution, the feature map is input into the next convolutional layer or pooling layer for high-dimensional feature extraction or feature dimensionality reduction. The dimensionality reduction methods of the pooling layer include max pooling and average pooling. Finally, the image's class is obtained at the fully connected layer. The architecture of CNN is illustrated in Figure 1.

2.2. Parameterized Quantum Circuits. PQC is a quantum computing model, and it can be run on quantum computers. PQC usually consists of parametric unitary gates and nonparametric unitary gates. The parametric gate includes single-qubit rotation gate, and the three single-qubit rotation gates are defined as follows:

$$\begin{aligned}
R_x(\theta) &= \begin{bmatrix} \cos\left(\dfrac{\theta}{2}\right) & -i\,\sin\left(\dfrac{\theta}{2}\right) \\ -i\,\sin\left(\dfrac{\theta}{2}\right) & \cos\left(\dfrac{\theta}{2}\right) \end{bmatrix}, \\
R_y(\theta) &= \begin{bmatrix} \cos\left(\dfrac{\theta}{2}\right) & -\sin\left(\dfrac{\theta}{2}\right) \\ \sin\left(\dfrac{\theta}{2}\right) & \cos\left(\dfrac{\theta}{2}\right) \end{bmatrix}, \\
R_z(\theta) &= \begin{bmatrix} e^{-i(\theta/2)} & 0 \\ 0 & e^{i(\theta/2)} \end{bmatrix}.
\end{aligned} \tag{2}$$

According to the quantum circuit $U(\mathbf{w})$, the parameterized circuit performs the unitary transformation of quantum state $|x\rangle$ to produce the output quantum state $|y\rangle$, that is,

$$|y\rangle = U(\mathbf{w})|x\rangle, \tag{3}$$

where $\mathbf{w}$ is the parameter in the circuit, such as the angle of the qubit rotation gate. PQC allows great freedom to design quantum algorithms, and it can realize different quantum algorithms by combining different quantum gates. The parameters of PQC are updated using the quantum-classical training method, as illustrated in Figure 2.

The quantum computer performs unitary transformation on the input quantum state according to the quantum circuits and then measures the output quantum state. The measurement results are input into a classical computer to

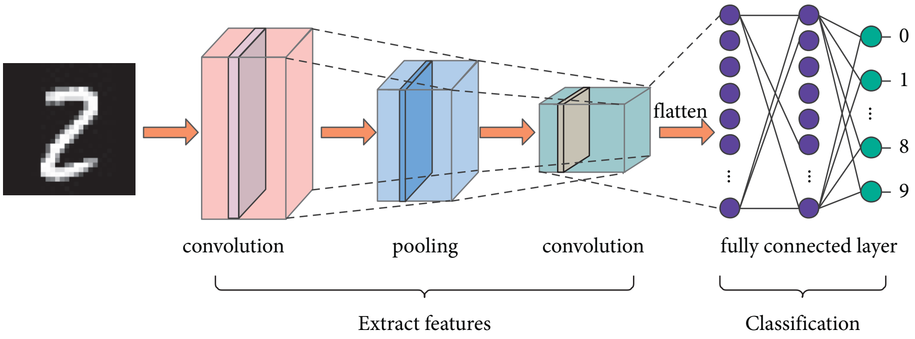


FIGURE 1: The architecture of CNN.

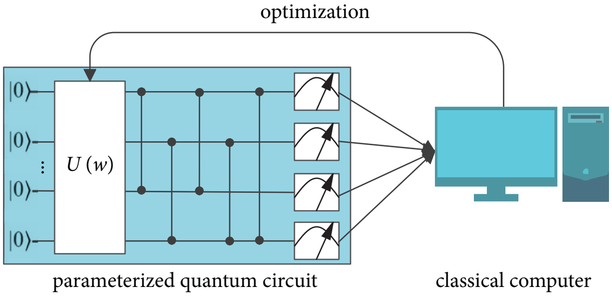


FIGURE 2: Schematic diagram of the parameterized quantum circuit.

calculate the loss value. The quantum circuit parameters are optimized through multiple iterations of the classical computer until the expectation value is satisfied or the maximum number of iterations is reached.

3. HQCCNN

3.1. Preparing the Quantum State. When the number of qubits in a quantum system is limited and the quantum system is used to tackle classical problems, the dimensionality reduction processing of classical data is usually required first. This paper downsamples the image to $m \times m$ size, scales the pixel value to $[0, 1]$, and then flattens the image.

We get a $1 \times m^2$ vector $\mathbf{x} = [x_1, x_2 \ldots x_{m^2}]$ after these operations and convert the vector to angle information α, that is,

$$\boldsymbol{\alpha} = \pi \mathbf{x}, \tag{4}$$

where $\alpha = [\alpha_1, \alpha_2 \ldots \alpha_{m^2}]$. The angle information α is taken as the rotation angle of the rotation gateR_y, and the rotation gate R_y is applied to the initial quantum state for encoding, for a m^2-input quantum system:

$$\left|\varphi_{img}\right\rangle = \otimes_{i=1}^{m^2} R_y(\boldsymbol{\alpha}_i)|0\rangle_1 \ldots |0\rangle_{m^2}. \tag{5}$$

Hence, $m \times m$ qubits are required to encode a downsampled image of size $m \times m$, and the quantum state $|\varphi_{img}\rangle$ is obtained by encoding all pixels of the downsampled image.

3.2. Quantum Convolutional Layer. After obtaining the quantum state $|\varphi_{img}\rangle$, the quantum convolution kernel $u(\theta)$ designed by PQC is used to perform unitary transformation on $|\varphi_{img}\rangle$. The convolution kernel has 5 training parameters $(\theta = \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$, as shown in the circuit structure in the purple window in Figure 3.

The circuit in Figure 3 shows a convolutional layer circuit with an image size of 2×3. We use the convolution kernel to perform unitary transformation on the qubits corresponding to the convolution window. It should be noted that the quantum convolution window is consistent with the classical convolution window, but it corresponds to 4 qubits. The purpose of the convolution window repeatedly acting on the 4 qubits is to preserve the characteristics of the convolution as much as possible and extract the hidden information from the quantum state.

3.3. Quantum Pooling Layer. In this study, the quantum pooling unitary gate V composed of 3 CNOT gates is used to reduce the dimension of the convolution results, as shown in the brown square in Figure 4. The pooling window acts in the same position as the convolution window, and after pooling, the convolution result of a convolution window is mapped to a qubit. Therefore, we only measure the specific qubit to obtain the expected value.

Classical CNN introduces nonlinearity through the nonlinear functions. In a quantum system, we introduce nonlinearity through measurement. After the evolution of the quantum system to the quantum state, the final quantum state $|\varphi_{\text{out}}\rangle$ is obtained. We perform Z-based measurement on the state $|\varphi_{\text{out}}\rangle$ to obtain the expectation value, that is,

$$E = \left\langle \varphi_{img} \middle| U^{\dagger}(\theta) V^{\dagger}(Z_1, \ldots, Z_N) V U(\theta) \middle| \varphi_{img} \right\rangle, \tag{6}$$

where $(Z_1, \ldots, Z_N)$ is a vector of Z operators acting on different qubits, V is the parameter-free unitary gate in the pooling layer, $U(\theta) = u_1(\theta) u_2(\theta) \ldots u_l(\theta)$, l is the number of convolutions in a convolutional layer, where for an image of $m \times m$, $l = (m-1)^2$, and the pooling unit also performs $(m-1)^2$ times in a pooling layer. If we directly measure the output of the quantum convolutional layer, we get a quantum output E with dimension $1 \times m^2$; if we measure the output of the quantum pooling layer, we get a vector E with dimension $1 \times (m-1)^2$. E is a vector composed of Z expectation values of different qubits, which is not directly related to the image label, so the E should be input into the classical fully connected layer for further processing.

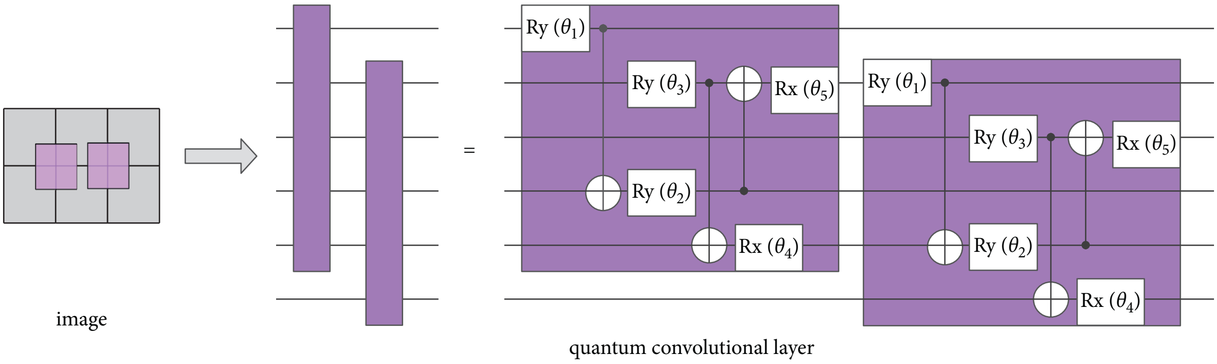


FIGURE 3: Convolutional circuit diagram.

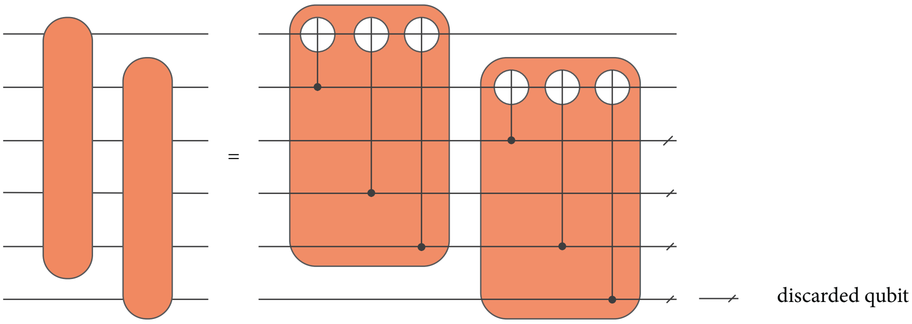


FIGURE 4: Pooling circuit diagram.

3.4. Generating Hybrid Network. HQCCNN consists of a quantum convolutional layer, a quantum pooling layer, and a classical fully connected layer. As shown in Figure 5, the quantum convolutional layer consists of multiple convolution kernels, which complete the quantum convolution to obtain the feature map. The convolution kernel in Figure 5 is the convolution structure introduced in Section 3.2, but the parameters of different convolution kernels are different. The convolution results are reduced by the quantum pooling layer, and the quantum pooling unit is the quantum pooling structure introduced in Section 3.3. Then, measuring the specific qubits, the measurement results are input into the fully connected layer to obtain the image's class.

3.5. Algorithm Execution Process. One of the reasons for the success of the classical neural network is that it has a developed backpropagation algorithm. We rely on the quantum computing framework TensorFlow Quantum [34] to complete the numerical simulation on a classical computer. TensorFlow Quantum implements a backpropagation algorithm that can be used in hybrid quantum-classical structures. We use the adjoint differentiation method to obtain gradients of the quantum system, which is described in detail in [34,35]. Through the abovementioned definition of the HQCCNN, the image classification algorithm is shown in Algorithm 1.

4. Simulation and Analysis

In this section, we use the MNIST dataset for simulation and analysis of HQCCNN. First, for each class in the MNIST dataset, 200 training images and 50 testing images are randomly selected. Therefore, in this paper, a total of 2000 images in the training set and 500 images in the testing set are used. However, the size of each image in the MNIST dataset is 28 × 28, which is not realistic for the current quantum experimental simulation, so we downsample each image to the size of 4 × 4.

The experimental setup in this section is as follows: first, we test all binary classification subsets in the MNIST dataset, then analyze the HQCCNN's structural changes on classification performance, and finally compare them with CNNs of the same architecture. And, all experiment details are set as follows: we use a classical hidden layer when performing binary classification tasks, and two hidden layers are used when performing multiple classification tasks. The hidden layers use ReLu nonlinear function, and the classic optimizer Adam [36] is used to optimize the parameters, the batch size is set to 32, the learning rate is 0.01, and 100 epochs are set.

4.1. Evaluation Indicators. To more intuitively evaluate the performance of HQCCNN, this study uses training set loss and testing set accuracy as evaluation indicators. Using the cross-entropy function as the loss function, the cross-entropy function is defined as follows:

$$H = -\frac{1}{M}\sum_{i}^{M}\mathbf{y}_i\log(\mathbf{p}_i), \tag{7}$$

where M is the size of the training set, $\mathbf{y}_i$ is the one-hot vector of the i-th sample in the training set, and $\mathbf{p}_i$ is the predicted probability vector by the model. Accuracy is defined as

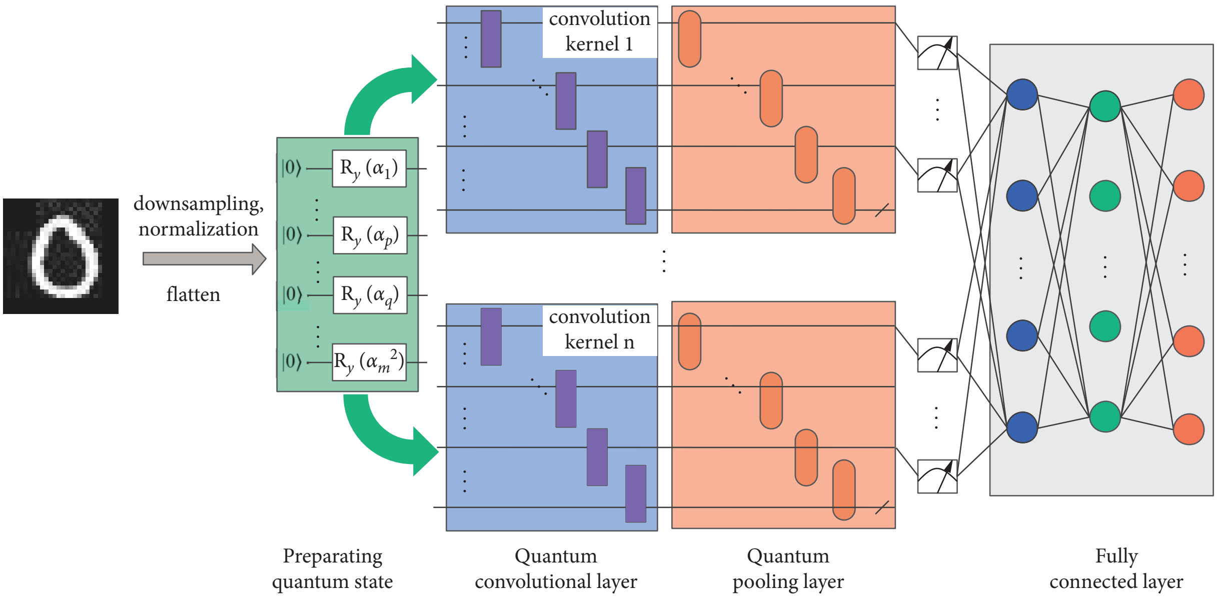


Figure 5: Hybrid quantum-classical convolutional neural network architecture.

```
Input: classical image dataset
preparing quantum state, the training set $|\varphi\rangle_{\text{train}}$ and the testing set $|\varphi\rangle_{\text{test}}$ are obtained.
constructing hybrid quantum-classical convolutional neural network
setting the values for batchsize, epoch, and learning rate
for each epoch do
    for each batchsize do
        quantum system perform unitary transformation $U(\theta)V|\varphi_{img}\rangle$,
        measuring, inputting E into the classical fully connected layer for further processing,
        calculating the loss value and gradient, updating the parameters based on the gradient and learning rate.
    end for
    validating testing set $|\varphi\rangle_{\text{test}}$
end for
Output: each image's class
```

Algorithm 1: HQCCNN classification algorithm.

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} (\arg\max p_i == y_i), \tag{8}$$

where N is the size of the testing set, y_i is the true class of the i-th sample, and the argmax function outputs the maximum value's position in a vector.

4.2. Binary Classification. In this section, we classify all binary subsets of MNIST and HQCCNN sets 1 convolutional layer, with 1 convolution kernel, and no pooling layer. The classical output layer uses a sigmoid nonlinear activation function. The binary classification results are shown in Figure 6.

Among all 45 binary classification tasks, HQCCNN shows high classification accuracy. The maximum appears in the classification task of digits 4 and 5, reaching 99.80%. The minimum value of 91.51% appears in the classification task of digits 3 and 8. It shows that HQCCNN is perfectly qualified for the binary classification task. However, it can also be seen in the classification results that when other digits are classified with 5 or 8, the probability of lower accuracy is higher. The reason for this difference may be that after image downsampling, some image features are lost, which increases the similarity between images.

Reference [37] proposes a quantum tensor network model. We compare some binary classification tasks with reference [37]. Reference [37] uses a model with 1008 parameters to test all binary classification tasks, while we use 150 parameters. The result is shown in Figure 7; it can be clearly seen from the figure that in the binary classification subsets \{0, 9\}, \{1, 7\}, \{2, 9\}, \{4, 6\}, \{6, 7\}, and \{8, 9\}, the accuracy of our model is better than in reference [37].

4.3. The Influence of Convolution Kernel on HQCCNN. Next, we discuss the influence of the convolution kernel's number on HQCCNN. 2 HQCCNNs with 1 quantum convolutional layer are set, and the number of convolution

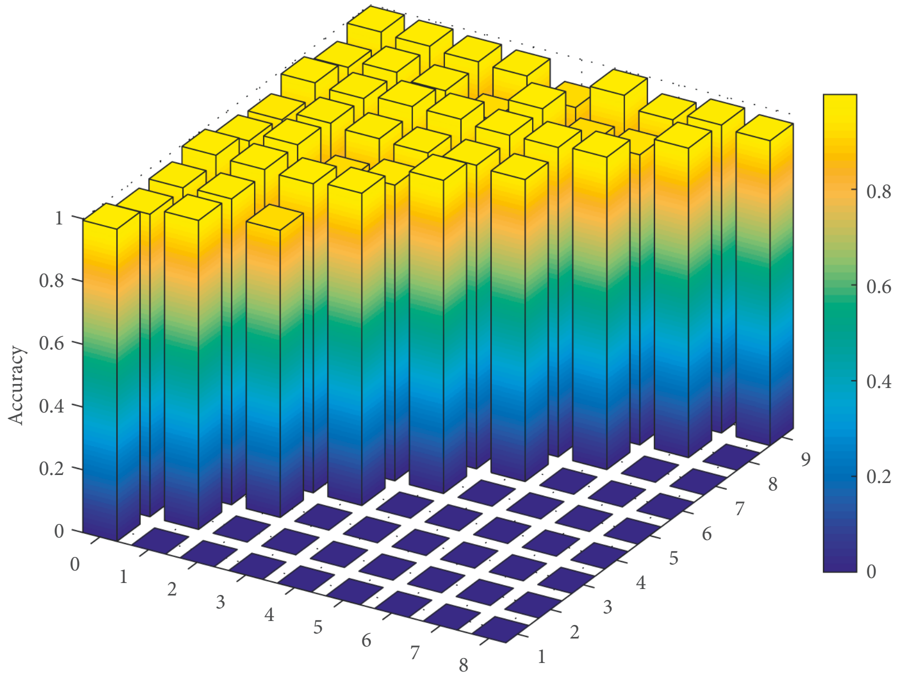


Figure 6: The binary classification results.

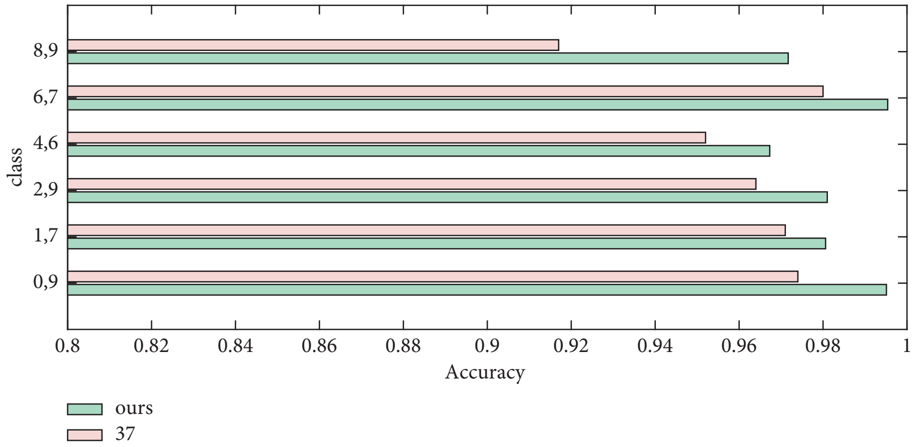


Figure 7: Comparison result with [37].

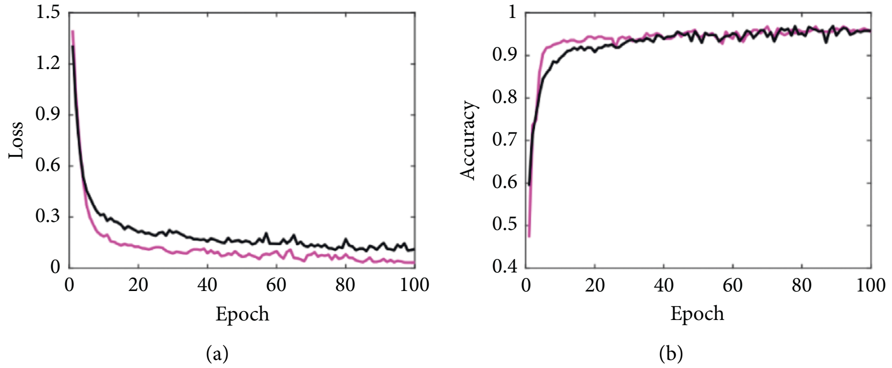
kernels is 2 and 4, respectively. The fully connected layers are the same, and the output layer uses the softmax nonlinear activation function. We set a multiclassification task of digits 0–4. The results are plotted in Figure 8.

According to Figure 8, the HQCCNN with 4 convolution kernels achieves a smaller training loss value, and at the beginning of the epoch, it achieves a higher classification accuracy than the HQCCNN with 2 convolution kernels. However, with the increase in the epoch, the accuracy of the 2 architectures is almost the same, indicating that the HQCCNN with 2 convolution kernels already has high learning ability and image classification ability for the 5-class classification task.

4.4. The Influence of Pooling Layer on HQCCNN. To explore the influence of adding a pooling layer on the performance of HQCCNN, we construct a HQCCNN with a pooling layer. HQCCNN sets 1 convolutional layer with 2 convolution kernels. After adding a pooling layer, the convolution results are pooled and then input into the same classical fully connected layer. We set a classification task of digits 0–4. The training set loss and testing set accuracy of HQCCNN are plotted in Figure 9.

When adding a pooling layer, compared to HQCCNN without adding a pooling layer, the classification performance of HQCCNN is degraded. However, it still maintains high performance, indicating that the pooling unit used in this paper can keep the model with high learning ability and image classification ability.

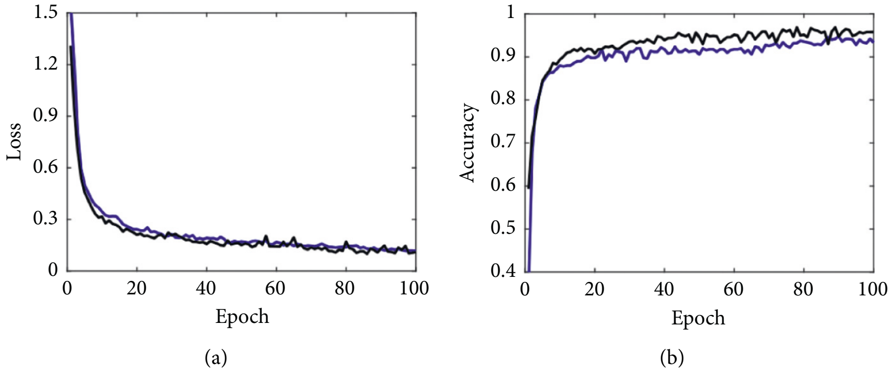
4.5. Comparative Experiment of HQCCNN and CNN. We explore the difference in classification performance between HQCCNN and CNN. We take the 5-class (0–4) task as an



(a)

(b)

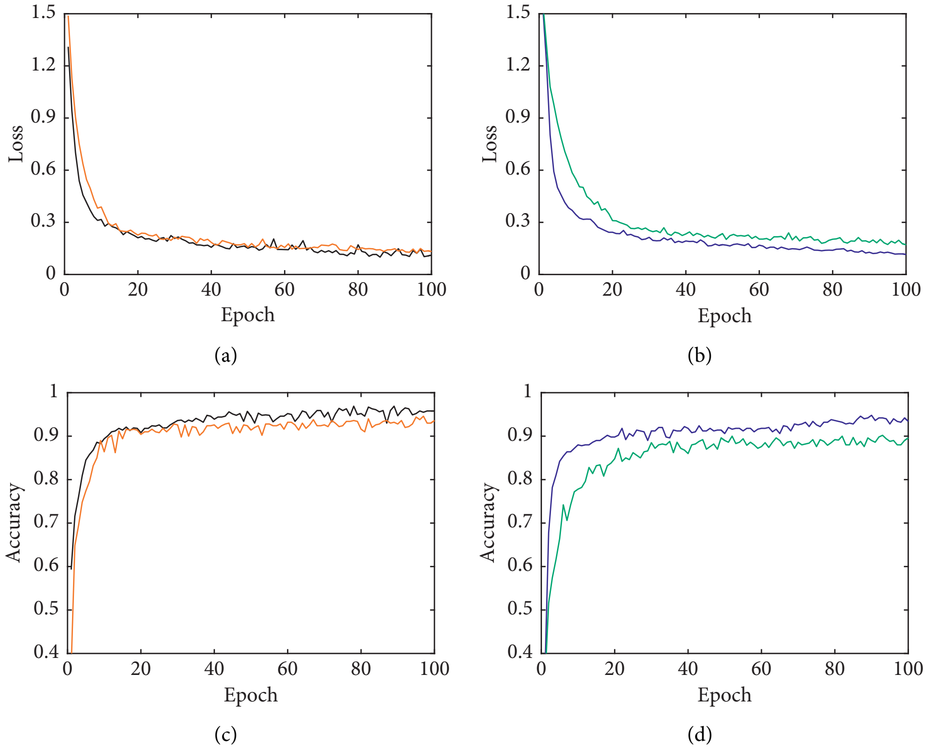
Figure 8: The results of HQCCNN. (a) The training set loss value and (b) the testing set accuracy. The black solid line represents the results of 2 convolution kernels, and the purple solid line represents 4 convolution kernels.



(a)

(b)

Figure 9: The results of HQCCNN with pooling. (a) The training set loss value and (b) the testing set accuracy. The black solid line is the result without adding pooling layer, and the blue solid line is the results with adding pooling layers.



(a)

(b)

(c)

(d)

Figure 10: The comparison results of HQCCNN and CNN. (a, c) The numerical results of HQCCNN and CNN without adding a pooling layer, where the black solid line is the result of HQCCNN and the yellow solid line is the result of CNN. (b, d) The numerical results of HQCCNN and CNN with pooling layers, where the blue solid line is the result of HQCCNN and the green solid line is the result of CNN.

Table 1: Complex binary classification task results.

Classification task	Model	Accuracy	
		HQCCNN	CNN
Even or odd	With pooling	0.8759	0.8645
	Without pooling	0.9240	0.9011
>4 or <4	With pooling	0.8931	0.8874
	Without pooling	0.9252	0.9066

example to illustrate the curve change in the numerical results of HQCCNN and CNN and set up other classification tasks to illustrate the performance differences. We set HQCCNN and CNN with the same network architecture, which means that the number of parameters between HQCCNN and CNN is the same. HQCCNN sets 1 convolutional layer with 2 convolution kernels and sets 2 architectures, one without pooling layer and another with a pooling layer. The corresponding CNN sets 1 convolutional layer with 2 convolution kernels, the convolutional layer with padding 1, the convolution kernel size is 2 × 2, with step size 1, and the activation function is ReLu. In the pooling layer, max pooling is used, and without padding, kernel size is 2 × 2, with step size 1. The classical fully connected layers are the same. The training set loss and testing set accuracy are plotted in Figure 10.

Next, we set up two complex binary classification tasks. The numerical results of HQCCNN and CNN are shown in Table 1.

As can be seen from Figure 10, without pooling layer, HQCCNN and CNN have almost the same loss value. However, HQCCNN has higher testing set accuracy than CNN. After adding a pooling layer, HQCCNN has a faster convergence speed and higher testing set accuracy than CNN. From Table 1, for complex binary classification tasks, HQCCNN also shows strong competitiveness. Figure 10 and Table 1 also show that the hybrid quantum-classical model has better learning ability and classification performance than CNN.

5. Conclusion

The development of quantum computing mode has made it possible to use quantum computing paradigm to solve classical problems. However, because of the limited number of available qubits in quantum computers at this stage, dealing with high-dimensional data is difficult. Thus, only the dimensionality of the data can be reduced, such as image downsampling. Although image downsampling is a compromise for current quantum devices, it makes it difficult to classify images because several images may become similar. Image downsampling makes the features disappear or become blur, thereby increasing uncertainty in the classification process. Amplitude encoding is a more efficient data-encoding approach, but it will result in intractable complexity. With researchers' efforts, the advent of large-scale quantum computers will facilitate the handling of high-dimensional data, which will additionally enable them to handle more classical problems.

In conclusion, we propose a hybrid quantum-classical convolutional computing model that includes two parts: a quantum computing part and a classical computing part. The quantum computing part includes quantum convolutional layer and quantum pooling layer. The quantum convolutional layer has a parameterized quantum circuit design to achieve the linear unitary transformation of the quantum state. The potential of quantum computing can be harnessed using parameterized quantum circuits. Classical computing is used to map the expectation value of the quantum system further, making it related to the image label. In the experimental simulation, we classify all the binary subsets of MNIST dataset and achieve better performance. Furthermore, we discuss the impact of model architecture changes on HQCCNN and conduct comparative experiments using CNNs with the same architecture. The results show that compared to CNN, HQCCNN has a faster training speed and higher testing set accuracy.

Data Availability

Data are available on request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61772295), Natural Science Foundation of Shandong Province, China(Grant Nos. ZR2021MF049 and ZR2019YQ01), and Project of Shandong Provincial Natural Science Foundation Joint Fund Application (ZR202108020011).

References

[1] S. S. Li, G. L. Long, F. S. Bai, S. L. Feng, and H. Z. Zheng, "Quantum computing," *Proceedings of the National Academy of Sciences*, vol. 98, no. 21, pp. 11847-11848, 2001.

[2] E. Knill, "Quantum computing," *Nature*, vol. 463, no. 7280, pp. 441–443, 2010.

[3] P. Xu, Z. He, T. Qiu, and H. Ma, "Quantum image processing algorithm using edge extraction based on Kirsch operator," *Optics Express*, vol. 28, no. 9, p. 12508, 2020.

[4] Y. Ma, N. Li, W. Zhang, S. Wang, and H. Ma, "Image encryption scheme based on alternate quantum walks and discrete cosine transform," *Optics Express*, vol. 29, no. 18, p. 28338, 2021.

[5] H. S. Zhong, H. Wang, Y. H. Deng et al., “Quantum computational advantage using photons,” *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020.

[6] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical Review Letters*, vol. 113, no. 13, Article ID 130503, 2014.

[7] S. Lu and S. L. Braunstein, “Quantum decision tree classifier,” *Quantum Information Processing*, vol. 13, no. 3, pp. 757–770, 2014.

[8] N. R. Zhou, X. X. Liu, Y. L. Chen, and N. S. Du, “Quantum K-Nearest-Neighbor image classification algorithm based on K-L transform,” *International Journal of Theoretical Physics*, vol. 60, no. 3, pp. 1209–1224, 2021.

[9] T. Xin, L. Che, C. Xi et al., “Experimental quantum principal component analysis via parametrized quantum circuits,” *Physical Review Letters*, vol. 126, no. 11, Article ID 110502, 2021.

[10] A. Peruzzo, J. McClean, P. Shadbolt et al., “A variational eigenvalue solver on a photonic quantum processor,” *Nature Communications*, vol. 5, no. 1, p. 4213, 2014.

[11] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, “The theory of variational hybrid quantum-classical algorithms,” *New Journal of Physics*, vol. 18, no. 2, Article ID 023023, 2016.

[12] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, “Circuit-centric quantum classifiers,” *Physical Review A*, vol. 101, no. 3, Article ID 032308, 2020.

[13] Y. Zeng, H. Wang, J. He, Q. Huang, and S. Chang, “A multi-classification hybrid quantum neural network using an all-qubit multi-observable measurement strategy,” *Entropy*, vol. 24, no. 3, p. 394, 2022.

[14] D. Wecker, M. B. Hastings, and M. Troyer, “Progress towards practical quantum variational algorithms,” *Physical Review A*, vol. 92, no. 4, Article ID 042303, 2015.

[15] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, “Variational quantum algorithms for discovering Hamiltonian spectra,” *Physical Review A*, vol. 99, no. 6, Article ID 062304, 2019.

[16] J. Xiao, J. Wen, S. Wei, and G. Long, “Reconstructing unknown quantum states using variational layerwise method,” *Frontiers of Physics*, vol. 17, no. 5, Article ID 51501, 2022.

[17] G. G. Guerreschi and A. Y. Matsuura, “QAOA for Max-Cut requires hundreds of qubits for quantum speed-up,” *Scientific Reports*, vol. 9, no. 1, p. 6903, 2019.

[18] L. Zhou, S. T. Wang, S. Choi, H. Pichler, and M. D. Lukin, “Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices,” *Physical Review X*, vol. 10, no. 2, Article ID 021067, 2020.

[19] S. E. Rasmussen, N. J. S. Loft, T. Bækkegaard, M. Kues, and N. T. Zinner, “Reducing the amount of single-qubit rotations in VQE and related algorithms,” *Advanced Quantum Technologies*, vol. 3, no. 12, Article ID 2000063, 2020.

[20] A. V. Uvarov, A. S. Kardashin, and J. D. Biamonte, “Machine learning phase transitions with a quantum processor,” *Physical Review A*, vol. 102, no. 1, Article ID 012415, 2020.

[21] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H. L. Huang, “Hybrid quantum-classical convolutional neural networks,” *Science China Physics, Mechanics & Astronomy*, vol. 64, no. 9, Article ID 290311, 2021.

[22] M. Schuld and N. Killoran, “Quantum machine learning in feature hilbert spaces,” *Physical Review Letters*, vol. 122, no. 4, Article ID 040504, 2019.

[23] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Reading text in the wild with convolutional neural networks,” *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[24] N. Sharma, V. Jain, and A. Mishra, “An analysis of convolutional neural networks for image classification,” *Procedia Computer Science*, vol. 132, pp. 377–384, 2018.

[25] P. M. Cheng and H. S. Malhi, “Transfer learning with convolutional neural networks for classification of abdominal ultrasound images,” *Journal of Digital Imaging*, vol. 30, no. 2, pp. 234–243, 2017.

[26] Q. Gao, S. Lim, and X. Jia, “Hyperspectral image classification using convolutional neural networks and multiple feature learning,” *Remote Sensing*, vol. 10, no. 2, p. 299, 2018.

[27] H. W. Wang, Y. J. Xue, Y. L. Ma, N. Hua, and H. Y. Ma, “Determination of quantum toric error correction code threshold using convolutional neural network decoders,” *Chinese Physics B*, vol. 31, no. 1, Article ID 010303, 2021.

[28] S. Liu, B. Tang, Q. Chen, and X. Wang, “Drug-drug interaction extraction via convolutional neural networks,” *Computational and Mathematical Methods in Medicine*, vol. 2016, Article ID 6918381, 8 pages, 2016.

[29] S. Wei, Y. Chen, Z. Zhou, and G. Long, “A quantum convolutional neural network on NISQ devices,” *AAPPS Bulletin*, vol. 32, no. 1, p. 2, 2022.

[30] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quanvolutional neural networks: powering image recognition with quantum circuits,” *Quantum Machine Intelligence*, vol. 2, no. 1, p. 2, 2020.

[31] I. Cong, S. Choi, and M. D. Lukin, “Quantum convolutional neural networks,” *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[32] Y. Li, R. G. Zhou, R. Xu, J. Luo, and W. Hu, “A quantum deep convolutional neural network for image recognition,” *Quantum Science and Technology*, vol. 5, no. 4, Article ID 044003, 2020.

[33] R. Parthasarathy and R. T. Bhowmik, “Quantum optical convolutional neural network: a novel image recognition framework for quantum computing,” *IEEE Access*, vol. 9, pp. 103337–103346, 2021.

[34] M. Broughton, G. Verdon, T. McCourt et al., “Tensorflow quantum: a software framework for quantum machine learning,” 2020, https://arxiv.org/abs/2003.02989.

[35] X. Z. Luo, J. G. Liu, P. Zhang, and L. Wang, “Yao.jl: extensible, efficient framework for quantum algorithm design,” *Quantum*, vol. 4, p. 341, 2020.

[36] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” 2014, https://arxiv.org/abs/1412.6980?context=cs#:%7E:text=We%20introduce%20Adam%2C%20an%20algorithm,estimates%20of%20lower%2Dorder%20moments.

[37] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, “Towards quantum machine learning with tensor networks,” *Quantum Science and technology*, vol. 4, no. 2, Article ID 024001, 2019.