IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 24, 2022, accepted February 9, 2022, date of publication February 16, 2022, date of current version February 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152213

Quantum Dilated Convolutional Neural Networks

YIXIONG CHEN
IBM Corporation, KIC Technology Center, Shanghai 200433, China
e-mail: chenyixiong516@msn.com

ABSTRACT In recent years, with rapid progress in the development of quantum technologies, quantum machine learning has attracted a lot of interest. In particular, a family of hybrid quantum-classical neural networks, consisting of classical and quantum elements, has been massively explored for the purpose of improving the performance of classical neural networks. In this paper, we propose a novel hybrid quantum-classical algorithm called quantum dilated convolutional neural networks (QDCNNs). Our method extends the concept of dilated convolution, which has been widely applied in modern deep learning algorithms, to the context of hybrid neural networks. The proposed QDCNNs are able to capture larger context during the quantum convolution process while reducing the computational cost. We perform empirical experiments on MNIST and Fashion-MNIST datasets for the task of image recognition and demonstrate that QDCNN models generally enjoy better performances in terms of both accuracy and computation efficiency compared to existing quantum convolutional neural networks (QCNNs).

INDEX TERMS Quantum-classical neural networks, quantum dilated convolution, parameterized quantum circuits.

I. INTRODUCTION

A. BACKGROUND

Convolutional neural networks (CNNs), proposed by Yann LeCun *et al.* [1] in 1989, are one of the most powerful algorithms in the context of deep learning. The main advantage of CNNs is that they use multiple feature extraction stages to automatically and accurately learn important features from the data without any human supervision. Due to this advantage, CNNs have been tremendously successful in a broad array of high-level computer vision problems, including image recognition [2]–[5], object detection [6]–[8], and image segmentation [9]–[11]. In recent years, with further developments in deep learning, CNNs have also demonstrated promising performances in other machine learning areas such as time series forecasting [12], [13], speech recognition [14], and recommendation systems [15].

Parallelly, with recent achievements in quantum technologies (e.g. noisy intermediate-scale quantum (NISQ) processors are currently available), the domain of quantum machine learning has attracted growing concerns and triggered an enormous amount of work. Quantum machine learning is a research area with the purpose of utilizing quantum mechanical effects such as superposition and entanglement to improve the performance of machine learning algorithms. Even though quantum machine learning is a new discipline, it has witnessed a number of successful quantum extensions to classical machine learning problems, including support vector machines [16], clustering [17], [18], and principal component analysis (PCA) [19].

Among quantum machine learning algorithms, quantum convolutional neural networks (QCNNs), also known as hybrid quantum-classical convolutional neural networks, are a family of variational quantum algorithms that have recently become a very active research field. The central idea of QCNNs is to construct a quantum convolutional layer within neural networks based on parameterized quantum circuits to estimate complex kernel functions in high dimensional Hilbert space. Inspired by CNNs, Liu *et al.* [20] proposed the first QCNN model and implemented it for image recognition. Afterwards, the QCNN model was investigated further in various work [21]–[26]. Recently, it has been demonstrated in [27] that QCNN models can also achieve promising results in speech recognition.

B. COMPUTATIONAL PROBLEM OF QCNN

Despite these successes, QCNNs suffer from computational bottlenecks, which make it time-consuming to train them. First, quantum operations applied on n-qubit quantum circuits require unitary matrices of size $2^n \times 2^n$ which will scale exponentially as the size of quantum circuits. Moreover, the calculation of gradients, due to the parameter-shift

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

rule [28], [29], results in more quantum circuit executions, when QCNNs are trained on a real quantum device. As an example, a quantum filter with p trainable parameters will add $2p$ more quantum circuit executions for each training sample to compute the required gradients. Although this problem can be mitigated when QCNNs are implemented on quantum simulators that support more efficient gradient computation methods such as backpropagation [30], [31] and adjoint differentiation method [32], it is inevitable for QCNNs to face another challenge. In CNNs, a convolutional layer, due to local connectivity, performs a large amount of element-wise matrix multiplication operations. For example, an output 100×120 feature map of a convolutional layer is obtained from $100 \times 120 = 12,000$ multiplication operations. The computational cost will increase significantly with the feature map size. Fortunately, this computational issue in CNNs can be handled by using vectorization techniques [33], [34]. QCNNs, as the counterparts of CNNs, have the same problem. However, unlike CNNs, most of current quantum devices, including quantum hardware and quantum simulators, do not support vectorization. Despite the availability of more mature quantum devices in the NISQ era, executing a large number of quantum circuits would be impractical in general.

C. PREVIOUS WORKS

A few works have been done to investigate how to reduce the runtime complexity of QCNNs. In the first family of works, a small number of qubits required for the quantum circuit is achieved by using classical data preprocessing techniques to reduce the dimension of the input features fed into the quantum (convolutional) layer. For instance, Pramanik *et al.* [35] employ PCA to reduce the VGG-16 features for the variational quantum classifier (VQC), while Hur *et al.* [36] adopt autoencoding (AutoEnc) for the dimensionality reduction. Nevertheless, the performance of the model trained in this manner is likely to be compromised by the limited expressive power of the reduced features, as shown in [35]. The second family of works focuses on how to efficiently encode classical data into quantum states. Schuld and Killoran [37] propose and implement the *amplitude encoding* for variational quantum circuits, which is explored further in [38] for Flexible Representation of Quantum Images (FRQI). This type of encoding method is efficient in terms of the required qubits for data encoding but it relies on very deep quantum circuits which are unpractical on NISQ devices. In a different direction, some recent researchers [26], [39], [40] propose *angle encoding* (also referred to as *qubit encoding*) and its variants (e.g. *dense angle encoding*) which use a constant quantum circuit depth for state preparation. This encoding scheme requires one qubit to encode one or a limited number of components of the input feature vector and thus is not efficient for high-dimensional input features from a resource perspective. To trade off these two encoding methods mentioned above, Hur *et al.* [36] further develop a hybrid encoding approach which requires fewer number of qubits than the angle encoding and uses a shallower quantum circuit depth than the amplitude encoding. Moreover, Henderson *et al.* [22] employ a threshold-based encoding technique to reduce the input-state space and make it possible to obtain the output feature map through a look-up table during the quantum convolution process without needing to execute the same quantum circuit repeatedly on image segments. This method is easy to implement, but it is infeasible on real quantum devices, as mentioned in [22].

D. OUR WORK

Having reviewed all these challenges and developments, in this work, we propose a novel hybrid quantum-classical architecture which we will call *quantum dilated convolutional neural network* (QDCNN). Our approach, motivated by the dilated convolution in deep learning, is an extension of the architectures presented in [20] and [22], and helps reduce the computational cost of QCNNs in a different way compared to the aforementioned approaches. Dilated convolution, also known as atrous convolution, was originally developed for efficiently computing the undecimated discrete wavelet transform [41]. In recent years, dilated convolution has attracted more and more attention, and is widely used in semantic segmentation [11], [42]–[46]. Following these successes, dilated convolution has also been adopted for a broader set of tasks, such as object localization [47], time series forecasting [12], [13], and sound classification [48]. The advantage of dilated convolution is that it allows for effectively expanding the field of view of filters to capture larger context without increasing the number of parameters or the computational complexity. By virtue of dilated convolution, the proposed QDCNNs can generally improve the computational efficiency of existing QCNNs while achieving the better task performance.

In summary, the contributions of our work are

- We propose a novel architecture of quantum convolutional neural network based on quantum dilated convolution operation. To the best of our knowledge, our work is the first attempt to combine the concept of dilated convolution with variational quantum circuits.
- We conduct experiments using MNIST and Fashin-MNIST datasets and demonstrate the superior performance of QDCNN models over QCNN models.

II. MATERIALS AND METHODS

A. PRELIMINARIES

1) CONVOLUTION OPERATION

The convolutional layer, which performs an operation called a "convolution", plays a central role in the CNNs. In the context of convolutional networks, a convolution is a linear operation that involves the multiplication of a set of weights with the input. For a convolution operation, a *kernel* or *filter* is defined as a feature extractor which is a two-dimensional (2-D) array of learnable weights. A filter is applied to a filter-size patch of the input image called *receptive field* and a dot product is performed between the pixels within the receptive field and the weight values in the filter. Afterwards, the filter shifts to the next patch according to a step size called *stride*, and repeats the above process until it has swept across the entire image. The final output from the series of dot products between the filter weights and the values underneath the filter is called a *feature map*. Let us denote the output feature map by y and the input image by x. In the 2-D convolution process, the feature map y is obtained by applying a filter k to the input image x:

$$y[i,j] = \sum_q \sum_l k[q,l] \cdot x[i+q, j+l] \tag{1}$$

where i and j are location indices of y. The output feature map, due to the convolution operation, usually has lower spatial resolution than the input image. This reduction in dimensions can be avoided by employing *zero padding* technique, namely adding a border of pixels with value zero around the edges of the input image before the application of a filter. A hyper-parameter called *padding* can be defined to determine how many zero values to add to the border of the image. Generally, the spatial resolution o_w and o_h of the resulting feature map, extracted from an $i_w \times i_h$ input image by an $m \times n$ kernel, can be calculated as:

$$o_w = \left(\frac{i_w - m + 2p}{s}\right) + 1 \tag{2}$$

$$o_h = \left(\frac{i_h - n + 2p}{s}\right) + 1 \tag{3}$$

where p and s represent the padding and stride respectively.

2) DILATED CONVOLUTION

Dilated convolution is a type of convolution that expands the kernel by inserting holes (i.e. points with weight of zero) between the consecutive kernel elements. In simple terms, dilated convolution is just a convolution applied to the input with defined gaps. Compared to standard convolution, dilated convolution introduces an extra hyperparameter called *dilation rate* that determines the stride with which the input pixels are sampled. According to the definition of dilated convolution, $r - 1$ zero values are inserted between two consecutive filter values, if the dilation rate is denoted by r. In this spirit, (1) needs to be reformulated as follows:

$$y[i,j] = \sum_q \sum_l k[q,l] \cdot x[i+q\cdot r, j+l\cdot r] \tag{4}$$

in the context of dilated convolution. It can be seen from (4) that dilated convolution is able to capture a larger receptive field without introducing more learnable parameters compared to standard convolution with the same kernel size. Moreover, for dilated convolution, we also need to rewrite (2) and (3) as:

$$o_w = \left(\frac{i_w - m - (m-1)(r-1) + 2p}{s}\right) + 1 \tag{5}$$

$$o_h = \left(\frac{i_h - n - (n-1)(r-1) + 2p}{s}\right) + 1 \tag{6}$$

which indicate that dilated convolution generally results in a feature map of a smaller size compared to standard convolution for the same set of hyperparameters. It is worth noting that the standard convolution can be regarded as a special case of dilated convolution with dilation rate $r = 1$.

3) QUANTUM CONVOLUTION

In contrast to classical convolution, quantum convolution is a new type of convolution based on quantum circuits and it generally consists of three modules:

- *ENCODING MODULE*. In this module, classical data are encoded into a quantum state which is further processed in the quantum convolutional circuit. There exist various encoding methods such as angle encoding, amplitude encoding and basis encoding. A summary of them can be found in the literature [49]. Among these methods, angle encoding is the most commonly used encoding approach. In this encoding scheme, the classical input is treated as the rotation angle of a single-qubit rotation gate (e.g. an RY rotation gate). For example, a classical variable or feature a can be encoded by $RY(a)$ which is applied on some initial state (e.g. vacuum state $|0\rangle$). In this sense, we can say that the classical information a is encoded into the initial state of a qubit. This type of angle encoding is called *one variable/qubit encoding*. This approach requires n qubits to encode n input variables. To reduce the required qubits, we can also encode multiple variables by sequential rotations applied on a single qubit. For example, input variables a_1, a_2, and a_3 can be encoded using $RX(a_1)$, $RY(a_2)$, and $RZ(a_3)$ rotation gates applied successively on a single qubit. This angle encoding is called *multiple variables/qubit encoding* or *dense angle encoding*. In this paper, we focus on one variable/qubit encoding method. Let us denote by $E(x)$ the encoding operator where x is the input vector. Then the encoded quantum state is obtained by:

$$|x\rangle = E(x)|0\rangle. \tag{7}$$

  It is worth noting that $E(x)$ usually contains the Hadamard gate which transforms the initial state into a uniform superposition state.
- *ENTANGLEMENT MODULE*. In this module, a cluster of single- and multi-qubit gates are applied to the encoded quantum state obtained from the previous module. Multi-qubit gates are usually *CNOT* gates and parametric controlled rotation gates (e.g. $CRZ(\theta)$ where θ is a trainable parameter), and they are used to generate correlated quantum states, namely entangled states. Single-qubit gates are mainly parametric rotation gates. This combination of single- and multi-qubit gates is referred to as a parameterized layer in the quantum convolution and is designed to extract task-specific features. This parameterized layer is usually repeated multiple times to extend the feature space. If we denote all unitary operations in the entanglement module by $U(\theta)$ for simplicity, the output quantum state will be:

$$|x,\theta\rangle = U(\theta)|x\rangle. \tag{8}$$

- *DECODING MODULE*. At this stage, certain local observable $A^{\otimes m}$ (e.g. Pauli Z operator $\sigma_z^{\otimes m}$) is measured in the final quantum state $|x,\theta\rangle$ from the entanglement module, where m is equal to or smaller than the total number of qubits n in the quantum system. The expectation value of the chosen observable $A^{\otimes m}$ can be obtained through repeated measurements:

$$f(x,\theta) = \langle x,\theta|A^{\otimes m}|x,\theta\rangle \tag{9}$$

So the purpose of this layer is to extract a classical output vector $f(x,\theta)$ by using the mapping from the quantum state to a classical vector:

$$\mathcal{M}: \quad |x,\theta\rangle \rightarrow f(x,\theta). \tag{10}$$

This classical vector $f(x,\theta)$ can be used as the input features for the subsequent layer in the QCNN.

B. QDCNN

The proposed QDCNN is designed in the same fashion as QCNNs described in literatures [20], [22]. Our model integrates quantum layers with classical layers and the quantum circuit ansatz can be placed anywhere in the model (e.g. at the beginning of the network, at intermediate layers in the network).

The key difference between our method and existing QCNNs is that the dilated convolution is employed for the quantum convolutional layer. So the quantum layer in QDCNNs is called *quantum dilated convolutional (QDC) layer*. An example of a QDC layer is illustrated in Fig. 1. Due to the mechanism of dilated convolution, the quantum kernel in our model generally covers larger image patches (i.e. receptive fields). For example, a 2 × 2 quantum dilated convolution with a dilation rate of 3 has a receptive field of 4 × 4 whereas the standard quantum convolution with the same kernel size has a receptive field of only 2 × 2. It is noteworthy that even though the quantum dilated convolution is able to expand the receptive field the number of data points that are fed into the quantum convolution circuit is the same as the one for the standard quantum convolution. This means that the quantum dilated convolution does not require more qubits than the standard quantum convolution with the same kernel size.

Our QDCNN model has mainly two advantages. Firstly, the QDC layer in our model, thanks to the enlarged receptive field, requires less number of times that the quantum kernel slides across the image (if there is no padding and the stride is the same), compared to the existing QCNN models. This can be understood by comparing (2) and (3) with (5) and (6) respectively. Therefore, using the QDC layer helps reduce the number of quantum circuit executions during the quantum convolution process. In the NISQ era, long training time is one of the biggest challenges facing the QCNN models. This difficulty stems mainly from the large number of quantum circuit executions from the quantum layers. In the quantum feature mapping process, due to the probabilistic characteristics, quantum measurement is usually performed multiple times (e.g. 1024) to get expectation values of some observables which can be considered as the extracted quantum feature maps. So how to reduce the number of quantum circuit executions plays a crucial role in mitigating the long-running-time problem of QCNNs. Our proposed quantum dilation is a powerful tool to explicitly control the amount of quantum circuit executions in the quantum layer.

The second advantage of our model is that it can improve the performance (e.g. classification accuracy) of the existing QCNN models. Due to the expanded receptive field, the QDC layer in our model generally reduces the spatial resolution of the resulting feature maps. However, these feature maps are extracted from larger receptive fields of the image and hence contain long-range context which plays an essential role in many machine learning tasks such as image recognition and image segmentation.

III. EXPERIMENTS

In this section, we conduct two experiments to evaluate the performance of our proposed QDCNN model and compare it with the existing QCNN model. In *Experiment A* and *Experiment B*, we construct quantum convolutional models with non-trainable and trainable quantum filters, respectively.

A. EXPERIMENT SETTINGS

1) DATASETS

We choose the image benchmark MNIST and Fashion-MNIST datasets [50], [51] for our experiments. The MNIST dataset contains 10 different classes of handwritten digits from ‘0’ to ‘9’, while the Fashion-MNIST dataset is a collection of 10 different shapes of t-shirts, dresses, shoes, etc. Both of these datasets have 60,000 training samples, and 10,000 test samples of 28 × 28 gray scale pixel images. Due to the expensive training and validation, we pick two subsets of the entire MNIST and Fashion-MNIST datasets, respectively, both of which consist of 1,000 balanced training samples and 200 balanced testing samples.

2) TESTED MODELS

In this research, we consider two types of models:

- *QDCNN Model.* We employ the architecture of the most basic convolution-inspired hybrid quantum-classical neural network. Our QDCNN model consists of one QDC layer with one filter and one fully-connected layer with 10 neurons. The kernel size and stride for the QDC layer is selected as 2 × 2 and 2 respectively without specification. The quantum circuit ansatz of the QDC layer is designed as follows. The 1 variable/qubit encoding scheme is adopted to encode the input image. Specifically, 2 × 2 pixels are encoded into a 4-qubit state using RY rotation gates. Note that, these 2 × 2 pixels are not adjacent to each other in the input image, due to the quantum dilated convolution. The resulting 4-qubit state is further transformed by a following random parameterized quantum circuit which might create the entanglement. The decoding method follows the same spirit of [52], in which each expectation value is mapped to a different channel of a single output pixel. Consequently, even though there is only one filter, the quantum layer can transform the input 2-D image into four feature maps. This type of quantum layer might benefit the model performance as it allows for correlation among

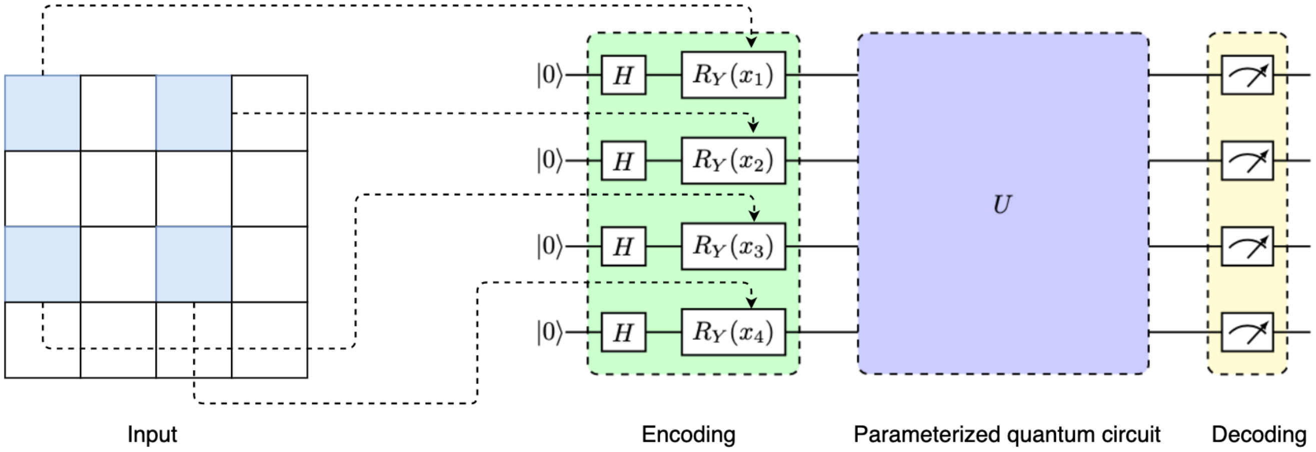

FIGURE 1. An example of a quantum dilated convolutional layer (QDC) with kernel size = 2 × 2 and dilation rate = 2. In contrast to standard quantum convolution, dilated convolution is applied to the input with defined gaps (one gap in this example) to enlarge the receptive field. Encoding, entanglement, and decoding modules for the QDC layer are highlighted with green, purple, and yellow colors respectively.

channels of the output feature maps. In both cases of MNIST and Fashion-MNIST datasets, the QDC layer extracts from the 28 × 28 input image a feature tensor of size 13 × 13 × 4, which is then transformed to 10 output probabilities by the fully-connected layer with softmax activation. To evaluate how the dilation rate impacts the model performance, we consider two QDCNN models with dilation rate $r = 2$ and $r = 3$, and refer to them as *QDCNN_r2* and *QDCNN_r3* respectively for the rest of the paper.

- *QCNN Model.* We select the standard QCNN model as our benchmark model. The QCNN model follows the same structure of our QDCNN model, with the only difference being that it uses a standard quantum kernel rather than a dilated one.

The random quantum circuit in each of these models consists of two 4-qubit random layers, each of which has four non-trainable or trainable parameters. For a fair comparison, all of these random circuits share the same architecture generated by the same random seed.

3) TRAINING SETUP

In Experiment A, after applying the non-trainable quantum filter to transform the original image data into feature maps, we use a mini-batch of 32 and Adam optimizer with a learning rate of 0.01 to train each model for 30 epochs. In Experiment B, due to the computational cost of training parametric quantum circuits involved in the trainable quantum filter, we reduce the batch size to four and train all models for 20 epochs with other hyperparameters remaining unchanged.

4) EXPERIMENTAL ENVIRONMENT

Experiments are conducted on the local computer with a 6-core CPU (2.2 GHz) by using PennyLane [53], Qulacs [54], and PyTorch [55]. PennyLane is an open source python-based framework that enables the automatic differentiation for hybrid quantum-classical computations. It is compatible with mainstream machine learning frameworks such as TensorFlow [56] and PyTorch, and it has a large plugin ecosystem which offers access to numerous quantum devices (i.e. simulators and hardware) from different vendors including IBM, Google, Microsoft, Rigetti, and QunaSys. In Experiment A, we perform the quantum processing of the original image data by using the *Qulacs* simulator [57] which is a high-performance C++ quantum simulator and made available through the community contributed PennyLane-Qulacs plugin [58]. In Experiment B, considering the large amount of quantum circuit executions required in the scheme of parameter-shift rule, we train all hybrid models by using instead the built-in Pennylane simulator *default.qubit* which supports the back-propagation method for the PyTorch interface.

B. RESULTS

As demonstrated in Table 1 and Table 2, our proposed QDCNN models exhibit significant performance benefits over the QCNN model in terms of run-time efficiency. More specifically, QDCNN models in Experiment A and B speed up the model training process by up to 15.24% and 18.42% respectively, compared with the QCNN model (QDCNN_r2 and QDCNN_r3 have similar training time because their QDC layers output feature maps of the same size in both experiments). These speedups result from the reduced number of quantum circuit executions, as discussed in the subsection II-B. Take Experiment A as an example. In this experiment, 13 × 13 quantum circuits need to be executed for both the QDC layers with dilation rate = 2 and dilation rate = 3 while 14 × 14 quantum circuits for the standard quantum convolutional layer. This means that QDCNN_r2 and QDCNN_r3 require 27 fewer quantum circuit executions per image than QCNN models. Compared with Experiment A, it generally takes a much longer time to train hybrid models in Experiment B, even though the quantum filter in each of them has only eight trainable parameters coming from the random circuit. This is mainly due to the fact that PennyLane does not support vectorization for quantum circuit executions. Nevertheless, quantum dilated convolution can still help reduce the training time significantly in this case.

Furthermore, it can also be seen from Table 1 and Table 2 that our QDCNN models generally enjoy higher recognition accuracy than the QCNN model. In particular, QDCNN_r3 achieves the best performance with regard to both validation loss and accuracy across all tasks. In light of the QDC layer with a dilation rate of 3, QDCNN_r3 provides up to 31.74% lower validation loss and up to 3% higher validation accuracy, compared with the QCNN model. This observed model performance boost stems mainly from the contextual information at larger scales captured by the QDC layer.

TABLE 1. Results of Experiment A on MNIST and Fashion-MNIST datasets, reported for three hybrid-classical neural network models with non-trainable quantum filters. QDCNN_r2 **and** QDCNN_r3 **are hybrid models with one QDC layer with dilation rate *r* = 2 and *r* = 3, respectively.** QCNN **represents the hybrid model with one standard quantum convolutional layer or equivalently QDC layer with dilation rate *r* = 1.**

Dataset	Method	Test acc	Test loss	Running time
MNIST	QCNN	88.00%	0.4389	624.648s
	QDCNN_r2	88.50%	0.3858	**529.472**s
	QDCNN_r3	**91.00%**	**0.3466**	530.270s
Fashion-MNIST	QCNN	78.50%	0.6319	610.978s
	QDCNN_r2	80.00%	0.6354	526.060s
	QDCNN_r3	**81.00%**	**0.6031**	**525.182**s

TABLE 2. Results of Experiment B on MNIST and Fashion-MNIST datasets, reported for three hybrid-classical neural network models with trainable quantum filters.

Dataset	Method	Test acc	Test loss	Running time
MNIST	QCNN	86.50%	0.5341	7.420h
	QDCNN_r2	86.50%	0.5315	6.436h
	QDCNN_r3	**89.50%**	**0.3646**	**6.053**h
Fashion-MNIST	QCNN	79.50%	0.8923	7.443h
	QDCNN_r2	77.00%	1.1038	**6.222**h
	QDCNN_r3	**80.50%**	**0.8837**	6.372h

IV. DISCUSSION

The results in the previous section suggest that the proposed QDCNN model can not only help overcome the computational bottleneck of the QCNN model but also achieve the better model performance, which is consistent with our expectations. To reduce the computational overhead of QCNN models, most of the existing studies focus on either classical data preprocessing or reducing quantum circuit complexity (e.g. width, depth). However, our work is a departure from these approaches and we pursue a more efficient way of applying quantum circuits to process the input data. In contrast to [35] and [36], our method does not rely on feature dimensionality reduction and thus makes use of all the information of the original data to maintain the model performance. Moreover, our model does not require complex data encoding techniques (e.g. hybrid encoding proposed in [36]) and the simple 1 variable/qubit encoding is sufficient. Both of these two differences stem from the use of quantum dilated convolution in our model. The computational cost of quantum convolution can be controlled by choosing appropriate hyperparameters (e.g. kernel size and dilation rate) of the QDC layer.

In spite of the series of results achieved in this work, some limitations still need to be resolved in the future research. Firstly, we study the QDCNN model only for the task of image recognition. Classical dilated convolution has successful applications in many other machine learning tasks such as image segmentation, speech recognition, and time series forecasting. Thus, quantum dilated convolution should also be further explored in these areas. Secondly, our experiments are conducted with quantum simulators. It would be interesting to implement the QDCNN model on real (noisy) quantum computers and compare hardware results with simulator results.

V. CONCLUSION

In this work, we propose a novel quantum machine learning model, which adopts the idea of dilated convolution in deep learning to the quantum neural network. We show through empirical evidence that the proposed QDCNN model outperforms the recent QCNN method in terms of computation time and recognition accuracy. In particular, we find that the quantum dilated convolution with a larger dilation rate generally contributes to a better model performance. Dilated convolution has been extensively studied in the area of deep learning, but little work has been done to explore it in the context of quantum machine learning. Our work constitutes a first step in this direction. With the promising results on both MNIST and Fashion-MNIST datasets, our QDCNN approach deserves further investigation in the future.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pp. 580–587, 2014.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2015, pp. 234–241.

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[12] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," 2017, *arXiv:1703.04691*.

[13] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, Dec. 2020.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*.

[15] F. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose, and X. He, "A simple convolutional generative network for next item recommendation," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 582–590.

[16] R. Varatharajan, G. Manogaran, and M. K. Priyan, "A big data classification approach using LDA with an enhanced SVM method for ECG signals in cloud computing," *Multimedia Tools Appl.*, vol. 77, no. 8, pp. 10195–10215, Apr. 2018.

[17] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "Q-means: A quantum algorithm for unsupervised machine learning," 2018, *arXiv:1812.03584*.

[18] J. S. Otterbach, "Unsupervised machine learning on a hybrid quantum computer," 2017, *arXiv:1712.05771*.

[19] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Phys.*, vol. 10, pp. 631–633, Jul. 2013.

[20] J. Liu, K. Hui Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, "Hybrid quantum-classical convolutional neural networks," 2019, *arXiv:1911.02998*.

[21] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, 2019.

[22] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: Powering image recognition with quantum circuits," *Quantum Mach. Intell.*, vol. 2, no. 1, pp. 1–9, Jun. 2020.

[23] S. Oh, J. Choi, and J. Kim, "A tutorial on quantum convolutional neural networks (QCNN)," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 236–239.

[24] S. Yen-Chi Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," 2020, *arXiv:2012.12177*.

[25] E. H. Houssein, Z. Abohashima, M. Elhoseny, and W. M. Mohamed, "Hybrid quantum convolutional neural networks model for COVID-19 prediction using chest X-ray images," 2021, *arXiv:2102.06535*.

[26] M. Alam, S. Kundu, R. O. Topaloglu, and S. Ghosh, "Iccad special session paper: Quantum-classical hybrid machine learning for image classification," *arXiv preprint arXiv:2109.02862*, 2021.

[27] C.-H.-H. Yang, J. Qi, S. Y.-C. Chen, P.-Y. Chen, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6523–6527.

[28] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 3, Sep. 2018, Art. no. 032309.

[29] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Phys. Rev. A, Gen. Phys.*, vol. 99, no. 3, Mar. 2019, Art. no. 032331.

[30] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT*, vol. 16, pp. 146–160, Dec. 1976.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

[32] T. Jones and J. Gacon, "Efficient calculation of gradients in classical simulations of variational quantum algorithms," 2020, *arXiv:2009.02823*.

[33] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for document processing," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.*, 2006, pp. 1–5.

[34] J. Ren and L. Xu, "On vectorization of deep convolutional neural networks for vision tasks," in *Proc. Conf. Artif. Intell.*, Jul. 2015, pp. 1840–1846.

[35] S. Pramanik, M. Girish Chandra, C. V Sridhar, A. Kulkarni, P. Sahoo, V. C. D. Sharma, A. Paliwal, V. Navelkar, S. Poojary, P. Shah, and M. Nambiar, "A quantum-classical hybrid method for image classification and segmentation," 2021, *arXiv:2109.14431*.

[36] T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification," 2021, *arXiv:2108.00661*.

[37] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, no. 4, Feb. 2019, Art. no. 040504.

[38] D. Mattern, D. Martyniuk, H. Willems, F. Bergmann, and A. Paschke, "Variational quanvolutional neural networks with enhanced image encoding," 2021, *arXiv:2106.07327*.

[39] M. Schuld and F. Petruccione, *Supervised Learning With Quantum Computers*, vol. 17. Berlin, Germany: Springer, 2018.

[40] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 3, Sep. 2020, Art. no. 032420.

[41] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," *Wavelets, Time-Frequency Methods Phase Space*, vol. 1, p. 286, Jan. 1989.

[42] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, pp. 1–4, Oct. 2016.

[43] L.-C. Chen, G. Papandreou, I. Kokkinos, K. P. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *CoRR*, vol. abs/1412.7062, pp. 1–14, Dec. 2015.

[44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2016.

[45] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[46] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1442–1450.

[47] Y. Kudo and Y. Aoki, "Dilated convolutions for image classification and object localization," in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, 2017, pp. 452–455.

[48] Y. Chen, Q. Guo, X. Liang, J. Wang, and Y. Qian, "Environmental sound classification with dilated convolutions," *Appl. Acoust.*, vol. 148, pp. 123–132, Dec. 2019.

[49] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data encoding patterns for quantum computing," in *Proc. Conf. Pattern Lang. Prog.*, 2019, pp. 1–11.

[50] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[51] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[52] A. Mari. (2021). *Neural Networks-PennyLane*. [Online]. Available: https://pennylane.ai/qml/demos/tutorial_quanvolution.htmlQuanvolutional

[53] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. Sohaib Alam, S. Ahmed, J. Miguel Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. Jakob Meyer, Z. Niu, A. Száva, and N. Killoran, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018, *arXiv:1811.04968*.

[54] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya, T. Yamamoto, T. Yan, T. Kawakubo, Y. O. Nakagawa, Y. Ibe, Y. Zhang, H. Yamashita, H. Yoshimura, A. Hayashi, and K. Fujii, "Qulacs: A fast and versatile quantum circuit simulator for research purpose," 2020, *arXiv:2011.13524*.

[55] A. Paszke, S. Gross, F. Massa, and A. Lerer, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Vancouver, BS, Canada, Dec. 2019, pp. 8026–8037.

[56] M. Abadi, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th Symp. Oper. Syst. Design Implement.*, pp. 265–283, 2016.

[57] (2019). *Qunasys*. [Online]. Available: https://github.com/qulacs/qulacsQulacs," 2019.

[58] S. Oud. (2019). *PennyLane Qulacs Plugin*. [Online]. Available: https://github.com/soudy/pennylane-qulacs

YIXIONG CHEN received the B.S. degree in theoretical physics from Lanzhou University, Lanzhou, China, in 2008, and the M.S. and Ph.D. degrees in theoretical physics from King's College London, London, U.K., in 2010 and 2015, respectively. He is currently working as a Data Scientist at IBM Corporation. His main research interests include quantum computing, quantum machine learning, and deep learning.