frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 23 November 2022
DOI 10.3389/fphy.2022.1069985



OPEN ACCESS

EDITED BY
Xiao Yuan,
Peking University, China

REVIEWED BY
Yukun Zhang,
Peking University, China
Yiming Huang,
Peking University, China

*CORRESPONDENCE
Denis Bokhan,
denisbokhan@mail.ru
Aleksey K. Fedorov,
akf@rqc.ru

SPECIALTY SECTION
This article was submitted to Quantum Engineering and Technology, a section of the journal Frontiers in Physics

RECEIVED 14 October 2022
ACCEPTED 02 November 2022
PUBLISHED 23 November 2022

CITATION
Bokhan D, Mastiukova AS, Boev AS, Trubnikov DN and Fedorov AK (2022), Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. *Front. Phys.* 10:1069985.
doi: 10.3389/fphy.2022.1069985



# Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning

Denis Bokhan[1,2]*, Alena S. Mastiukova[2,3], Aleksey S. Boev[2], Dmitrii N. Trubnikov[1] and Aleksey K. Fedorov[2,3]*

[1]Laboratory of Molecular Beams, Physical Chemistry Division, Department of Chemistry, Lomonosov Moscow State University, Moscow, Russia, [2]Russian Quantum Center, Moscow, Russia, [3]National University of Science and Technology "MISIS", Moscow, Russia

Multiclass classification is of great interest for various applications, for example, it is a common task in computer vision, where one needs to categorize an image into three or more classes. Here we propose a quantum machine learning approach based on quantum convolutional neural networks for solving the multiclass classification problem. The corresponding learning procedure is implemented *via* TensorFlowQuantum as a hybrid quantum-classical (variational) model, where quantum output results are fed to the softmax activation function with the subsequent minimization of the cross entropy loss *via* optimizing the parameters of the quantum circuit. Our conceptional improvements here include a new model for a quantum perceptron and an optimized structure of the quantum circuit. We use the proposed approach to solve a 4-class classification problem for the case of the MNIST dataset using eight qubits for data encoding and four ancilla qubits; previous results have been obtained for 3-class classification problems. Our results show that the accuracy of our solution is similar to classical convolutional neural networks with comparable numbers of trainable parameters. We expect that our findings will provide a new step toward the use of quantum neural networks for solving relevant problems in the NISQ era and beyond.

KEYWORDS

**quantum learning, multinomial classification, parameterized quantum circuit, variational circuits, amplitude encoding**

## 1 Introduction

Quantum computing is now widely considered as a new paradigm for solving computational problems, which are believed to be intractable for classical computing devices [1–5]. The idea behind quantum computing is to use quantum physics phenomena [2], such as superposition and entanglement. Specifically, in the quantum gate-based model, quantum algorithms are implemented as a sequence of logical operations under the qubits (quantum analogs of classical bits), which comprise the corresponding quantum circuits terminated by qubit-selective measurements [3].

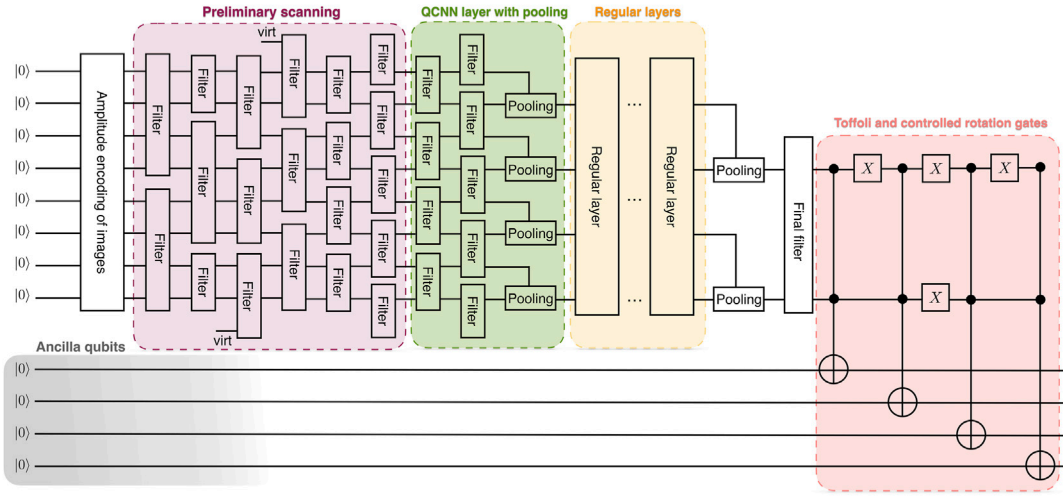


FIGURE 1
General structure of the proposed quantum neural network structure consisting of several steps: Preliminary scanning using $n$-qubit filters, pooling, and regular layers.

Examples of the problems, for whose quantum speedups are expected to be exponential, are prime factorization [4] and simulating quantum systems [5], for example, modelling complex molecules and chemical reactions [6]. The amount of computing power for such applications, however, significantly exceeds the resources of currently available quantum computing devices. For example, factoring RSA-2048 bit key requires 20 million noisy qubits [7], whereas currently available noisy intermediate-scale quantum (NISQ) devices have about 50–100 qubits [8]. Quantum computing can be also considered in the context of data processing [9] and machine learning applications [10], where the required resources for solving practical problems are expected to be not so high. Still the caveats of quantum machine learning are related to the input/output problems [11]: Although quantum algorithms can provide sizable speedups for processing data, they do not provide advantages in reading classical input data. The cost of reading the input then may in some cases dominate over the advantage of quantum algorithms. One may note that various approaches have been suggested, specifically, amplitude encoding [12], but the problem of the conversion of classical data into quantum data in the general case remains open [11].

The use of NISQ devices in the context of the quantum-classical (variational) model has emerged as a leading strategy for their use in the NISQ era [13, 14]. In such a framework, a classical optimizer is used to train a parameterized quantum circuit [13]. This helps to address constraints of the current NISQ devices, specifically, limited numbers of qubits and noise processes limiting circuit depths. An interesting link between the quantum-classical (variational) model and architectures of artificial neural networks opens up prospects for the use of such an approach for machine learning problems [15–22]. The workflow of variational quantum algorithms, where parameters of circuit are iteratively updated (optimized), resembles classical learning procedures [19].

A cornerstone problem of various machine-learning-based approaches is classification, that it why it has been widely considered from the view point of potential speedups using quantum computing. As it has been demonstrated in Refs. [9, 23], kernel-based quantum algorithms may provide efficient solutions for the classification problem. Specifically, the quantum version of the support vector machine [9] can be used as an optimized binary classifier with complexity logarithmic in the size of the vectors and the number of training examples. A distant-based quantum binary classification has been proposed in Ref. [24]. Alternative versions of binary quantum classifiers have been considered in Refs. [25–29] (for a review, see also Ref. [30]). A natural next step is to consider the multiclass classification, which has been addressed recently in Ref. [31] with the demonstration of the performance on the IBMQX quantum computing platform. This method uses single-qubit encoding and amplitude encoding with embedding of data, so the obtained results are of quite high accuracy for the 3-class classification task. Very recently, an approach based on quantum convolutional neural network (QCNN) [32] has been used for binary classification and a method to extend it to the multiclass classification case has been discussed. We also note that some of the proposed quantum machine learning algorithms have been tested in practically relevant settings, for example, analyzing NMR readings [33, 34] with the trapped-ion quantum computer, learning for the classification of lung cancer patients [35] and classifying and ranking

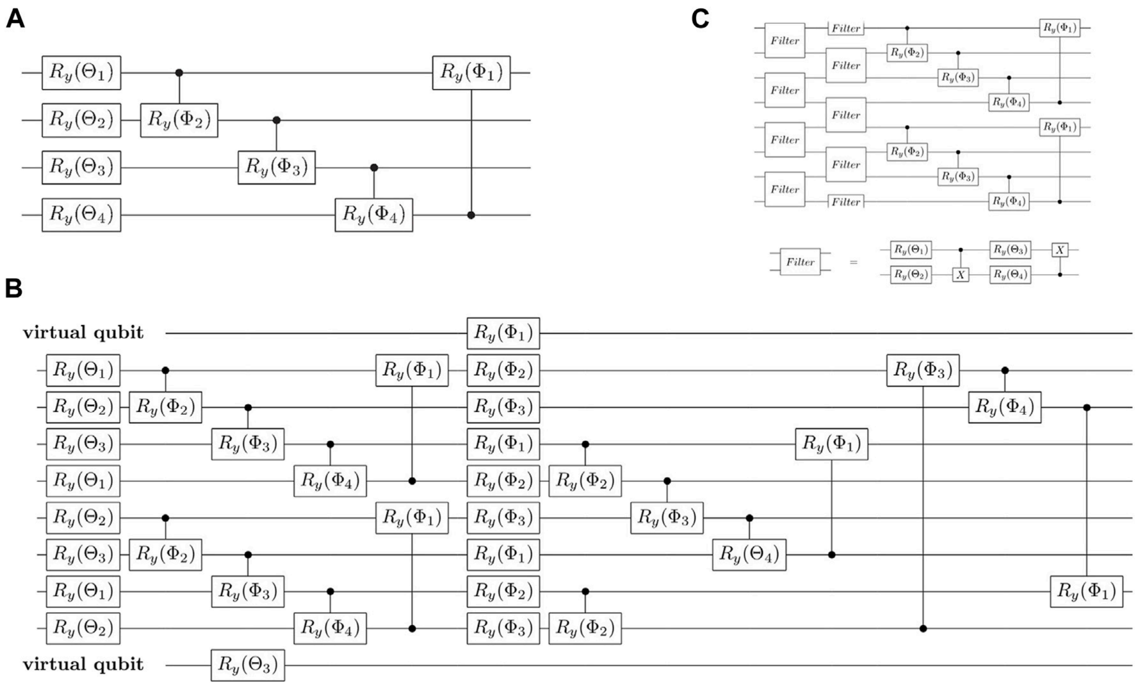

FIGURE 2
Quantum circuits for preliminary scanning: In **(A)** the 4-qubit filter with 4-qubit entanglement is shown; in **(B)** the stack of 3-qubit filters with 4-qubit entanglement is presented; in **(C)** the 2-qubit filters with 4-qubit entanglement is demonstrated.

DNA to RNA transcription factors [36] using a quantum annealer, weather forecasting [37] on the basis of the superconducting quantum computer, and many others [38].

In this work, we present a quantum multiclass classifier that is based on the QCNN architecture. The developed approach uses traditional convolutional neural networks, in which few fully connected layers are placed after several convolutional layers. The corresponding learning procedure is implemented *via* TensorFlowQuantum [39] as a hybrid quantum-classical (variational) model, where quantum output results are fed to softmax cost function with subsequent minimization of it *via* optimization of parameters of quantum circuit. Then we discuss the modification of a quantum perceptron, which enables us to obtain highly accurate results using quantum circuits with a relatively small number of parameters. The obtained results demonstrate successful solving of the classification problem for the 4-classes of MNIST images.

Our paper is organized as follows. In Section 2, we present the general description of the proposed quantum algorithm that is used for multiclass classification. In Section 3, we provide in-detail discussion of the layer of the proposed quantum machine learning algorithm. In Section 4, we demonstrate the results of the implementation of the proposed algorithm for multiclass image classification for hand-written digits from MNIST and clothes images from fashion MNIST datasets. We conclude in Section 5.

## 2 General scheme

The core concept that we use here is the hybrid (variational or quantum-classical) approach (for a review, see Refs. [13, 14]). This approach uses parametrized (variational) quantum circuits, where the exact parameters of quantum gates within the circuit can be changed. The general structure of our variational circuit is shown in Figure 1. Below we describe the proposed approach for multiclass classification based on the classical-quantum approach.

At the first step, we realize an amplitude encoding of input data, in our case, MNIST images. In fact, due to the high cost of this step [11], we generate a set of encoding circuits, and store their parameters and structures in a memory, thus generating a quantum dataset. We consider MNIST images, which are rescaled from 28 by 28 to 16 by 16 pixels, and, thus, 8 qubits are needed. In terms of the corresponding qubit states, encoded images can be expressed as follows:

$$\Psi_k = \sum_{m=0}^{N} C_m^k |m\rangle, \tag{1}$$

where $k$ is the index of image and $|m\rangle$ is a qubit register of 8 qubits, which encode index $m$, and $N = 255$. Coefficients $C_m^k$ are equal to elements of normalized flatten vectors of images. In general, this approach enables us to pack vector of N double-precision numbers into $\log_2(N)$ qubits, and, thus, significantly reduce the size of processed data. It should be noted however that existing algorithms for amplitude encoding scale exponentially with N; further study is needed to overcome this problem.

We first employ the amplitude encoding procedure [12], where ancilla qubits are used for one-hot encoding of the class of target images. Preliminary analysis of encoded images is performed with 3 convolutional layers with the sizes of filters, equal to 4, 3, and 2, respectively. Each such layer consists of 2 sublayers that are needed to maintain translational invariance (at least, partially), and all the filters of the same size contain identical trainable parameters as is the case for classical convolutional neural networks (CCNN). We note that for filters with the size of 3 we need a virtual qubit, which is always set to zero; such a method is needed to fit the filter into 8 qubits in the translationally invariant manner. The convolutional layer with pooling is then placed after preliminary layers; at this step the first reduction of the required qubit number is realized.

As in the classical setup, several fully connected layers are added after convolutional layers (9 layers in our case). The further reduction of qubit numbers is realized after regular layers and subsequent pooling are done (in the same way as it is done after convolutional filters).

The final filter is needed for mixing the information from two parts of divided circuit. In the process of learning the output of final filter would contain the codes of classes: $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Output cascade contains four Toffoli gates, which activate the corresponding ancilla qubit; at the end of the quantum circuit we have one-hot encoded by ancillas class of image. Measurement results of ancilla qubits are passed to the *softmax* activation function. The categorial cross-entropy is then used as the cost function. The subsequent calculations of gradients of the cost function with respect to the parameters of gates are done using parameter shift rule New parameters of quantum gates obtained by the gradient descent step. The detailed structure of all layers is described below.

# 3 Structure of layers

Here we present the detailed description of the layers that are used in our quantum machine learning algorithm.

## 3.1 Preliminary scanning using $n$-qubit filters

The structure of 4-qubit filters is presented in Figure 2A. First of all, $R_Y(\Theta_1)$, $R_Y(\Theta_2)$, $R_Y(\Theta_3)$ and $R_Y(\Theta_4)$ rotations are added in order to rotate each of the four qubits separately. We propose to use controlled parameterized rotations $R_Y(\Phi)$ for the entanglement, which is an essential new element in the structure of quantum perceptron. We note that in Ref. [31] authors use the standard controlled X gates for this purpose. Here, as we demonstrate, the parameterized entanglement scheme provides higher accuracy of image classification due to the more flexible learning algorithm.

In classical machine learning, the linear perceptron is passed through a certain non-linear function, which is essential for the learning process. In the quantum case, instead of summations of neurons we use entanglement of qubits. The degree of entanglement is controlled by the parameters $\Phi$, which makes the learning process more flexible, and, thus, the classification procedure may become more accurate. In fact, many classical activation functions like *sigmoid* or tanh behave akin to switches, so their values change from 0 or from −1 to 1 in a certain region. In the quantum domain, we can switch from separable (nonentangled) to entangled states, that play the role of non-linearities in classical learning. So far, individual rotations, which are followed by the parameterized entanglement, can be considered as an analog of the perceptron with the non-linearity.

After 4-qubit scanning, smaller-scale filters are applied to analyze obtained quantum feature map in more details. The structure of layers with 3-qubit filters is presented in Figure 2B. In order to rotate 3 individual qubits $R_Y(\Theta_1)$, $R_Y(\Theta_2)$ and $R_Y(\Theta_3)$ gates are added. Similarly to the case of 4-qubit filters, individual rotations are performed by parameterized $R_Y$ gates. We note that even in the case of 3-qubit filters, we use the entanglement of 4 qubits. Even though the entanglement of 3 qubits looks more intuitive in this case, as we show below, the 4-qubit entanglement provides more accurate results on image recognition. More detailed scanning of images is performed by layer with 2-qubit filters; the corresponding circuit is given in Figure 2C (also see Ref. [32]). As in all previous cases, we use 4-qubit entanglement and the filter consists of 4 individual rotations with additional entanglement by CNOT gates. The idea of using 4-qubit entanglement is inspired by classical CNN, where generation of new feature maps is done by summation of the shared weights of previous feature maps and subsequent application of non-linearity.

## 3.2 Quantum convolutional neural network layer with pooling

After the preliminary scanning step, the obtained quantum state of 8 qubits contains encoding of feature maps. The role of the next layer (see Figure 3) is to analyze these maps in more detail and pick up the most important of them. The scheme of the layer is given in Figure 3B, where the convolutional filter is the same as in Figure 3A. We note that in the pooling circuit, controlled $R_Z$ rotation is activated if the first qubit is at state 1, while the controlled $R_X$ gate is used when the upper qubit is at state 0. This is conceptually similar to the structure proposed in Ref. [32].

## 3.3 Regular layers

Similarly to the CCNN case, several regular layers are placed after convolutional layers. In our case we add 8 layers, as shown in Figure 3A. In order to get more accurate results, the double entanglement is added after individual rotations. The second reduction of qubit number in the circuit is done by two pooling procedures as in the case of convolution layers. In order to obtain the required structure, we add a final filter at the end of the quantum

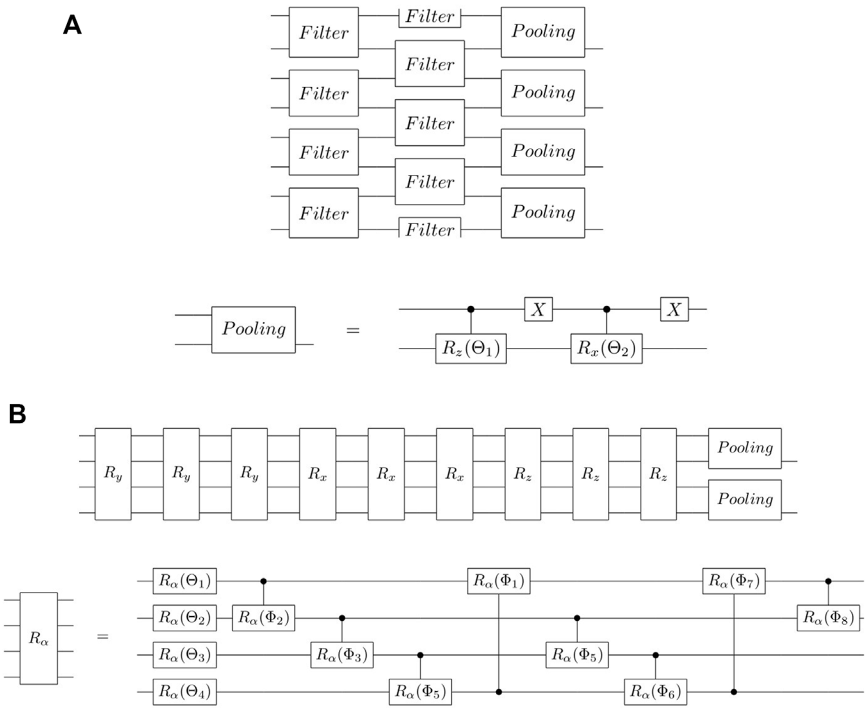


**FIGURE 3**
In **(A)** the convolutional layer with pooling is shown. In **(B)** the structure of regular layers is illustrated.

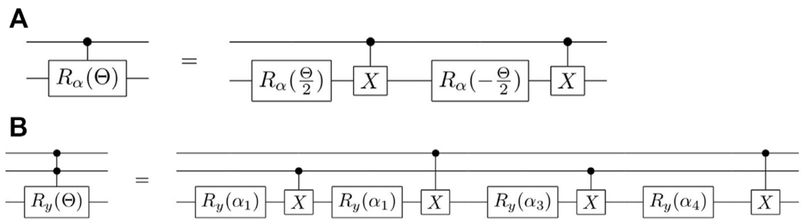


**FIGURE 4**
**(A)** Single-controlled rotation gates in terms of rotations and CNOT gates. **(B)** Decomposition of the Toffoli gate in terms of $R_Y$ rotations and CNOT gates.

circuit. As it is shown below, the use of the final filter is essential for obtaining more accurate results of image classifications.

## 3.4 Toffoli and controlled rotation gates

The practical realization of high-fidelity two-qubit operations on quantum hardware is still a challenging task. The situation is typically more difficult for three-qubit gates, such as a Toffoli gate. Thus, it is necessary to decompose these gates *via* single- and two-qubit gates, which can be practically performed. The general algorithm of $n$-controlled rotations is presented in Ref. [40] and for the case of single-controlled rotation it can be expressed as it is shown in Figure 4A. In order to implement the Toffoli gate, we consider the qubit inversion as a rotation operation around $X$ or $Y$ axes and in our case a doubly-controlled $R_Y(\Theta)$ gate is used with the value of $\Theta = 2\pi$. The circuit is presented in Figure 4B and it corresponds to the representation of the sum of parameterized $n$-controlled rotations, which are considered in Ref. [40]. A Toffoli gate, in fact, can be considered as a sum of such rotations with $n = 2$, where $\Theta$ angles of all rotations, except the one that is controlled by the 11th combination, are set to zero. The definition of $\alpha$ angles is realized along the lines of the previously described procedure [40]; they are obtained from $\Theta$ angles by simple matrix transformation.

We note that multiqubit gate decomposition can be further improved using qudits, which are multilevel quantum systems.

TABLE 1 Number of images of each type.

| MNIST digits | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Training | 5923 | 6742 | 5958 | 6132 | 5842 | 5421 | 5918 |
| Test | 980 | 1135 | 1032 | 1010 | 982 | 892 | 958 |
| **MNIST fashion** | **0** | **1** | **2** | **3** | **8** | **9** | |
| Training | 6000 | 6000 | 6000 | 6000 | 6000 | 6000 | |
| Test | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 | |

As it has been shown, the upper levels of qudits can be used instead of ancilla qubits in the decomposition [41–45].

## 4 Classification results

We benchmark the proposed quantum machine learning algorithm with the use of hand-written digits from MNIST and clothes images from fashion MNIST datasets. Examples are presented in Figure 5.

All the simulations are performed using Cirq python library for the constructions of quantum circuits; TensorFlowQuantum library [39] is used for the implementation of the machine learning algorithm with parameterized quantum circuits. We use the Adam version of gradient descent, with a learning rate equal to 0.00005, and the overall number of trainable parameters in the QCNN circuit is equal to 149. As a metric for the model performance, we simply use the accuracy of the recognition and for more detailed analysis two sets of experiments are done. In all the conducted experiments, parameterized quantum circuits are trained during 50 epochs.

Within the first set, training and classification are completed for the case, when the dataset consists of images which have certain similarities; thus, the classification problem becomes more difficult. We use MNIST images of digits 3, 4, 5, and 6 for this part. Also, fashion MNIST images with labels 0, 1, 2, and 3 are used for this purpose.

The second experimental set is focused on images, which strongly differ from one another thus making the recognition process easier; MNIST images of digits 0, 1, 2, and 3 and fashion MNIST images with labels 1, 2, 8, and 9 are used here. The total number of considered images of each type is given in Table 1.

Each image vector is normalized to one since only that type of vector can be used by the amplitude encoding algorithm. The results of image classification are given in Table 2. Quantum circuits for multiclass classification are considered in Ref. [31]. QCNN examples, provided within the documentation of TensorFlowQuantum [39] also can be relatively simply generalized for the case of multiclass classification tasks. In the second column of Table 2 we provide the results of experiments with circuits, similar to previous results [31]. In order to obtain these results we replace all the $R_Y(\Theta)$ used at entanglement steps by CNOT gates. Also, we remove all the parts of the circuit of Figure 1, which are placed after regular layers, i.e., pooling layers, final layers, and the part with Toffoli gates. Entanglement of ancilla qubits with regular layers is done via CNOT gates according to Figure 2 of Ref. [31]. The third column in Table 2 contains the results obtained with the full circuit shown in Figure 1. The significant improvement in the accuracy of the classification results is caused by two facts. Firstly, the usage of parameterized entanglement in our circuit. Secondly, the increased performance may be connected with the fact that our circuit is constructed in a similar way to classical neural networks—we use a qubit reduction procedure analogous to the reduction of the number of layer outputs in the classical case, i.e., reduction until the number of outputs equals the number of target classes. Note that in Figure 1 ancilla qubits are used only at the read-out step and no entanglement is needed between ancillas and other qubits during the computational procedure, significantly simplifying

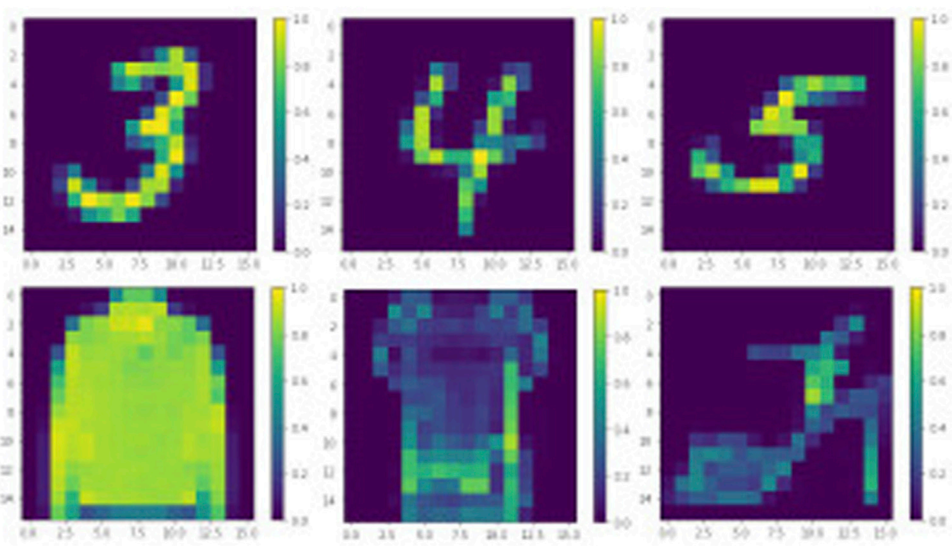

FIGURE 5
Examples of MNIST digits (top) and MNIST fashion (buttom) images used in experiments.

TABLE 2 Accuracies of classification for quantum and classical convolutional neural networks.

| | Quantum | Quantum | Classical |
|---|---|---|---|
| | Reference [31]—like | Figure 1 | |
| MNIST digits (3456) | 71.44 | 85.14 | 94.25 |
| MNIST digits (0123) | 77.64 | 90.03 | 95.85 |
| MNIST fashion (0123) | 71.15 | 85.93 | 89.69 |
| MNIST fashion (1289) | 79.33 | 93.63 | 97.42 |

TABLE 3 Structure of the used classical convolutional neural network with 188 parameters.

| Layer type | Output shape | Number of parameters |
|---|---|---|
| Conv2D | (None, 14, 14, 1) | 10 |
| Conv2D | (None, 12, 12, 1) | 10 |
| Pooling | (None, 6, 6, 1) | 0 |
| flatten | (None, 36) | 0 |
| Dense | (None, 4) | 148 |
| Dense | (None, 4) | 20 |

the corresponding quantum hardware requirements. We also compare obtained quantum results with results of the CCNN with a similar number of parameters, which is 188 in our case. The structure of the CCNN is presented in Table 3. Clearly, the classical results are more accurate, indicating that, with a similar number of parameters, the classical model is still more expressive. An analysis of possible quantum advantage in ML tasks is presented in Ref. [46]. In their study, the authors analyze ML models based on kernel functions and show that with enough data provided, classical methods become more powerful than the corresponding quantum algorithms. Thus, additional study is still needed to find ML tasks where quantum algorithms will outperform their classical analogs.

Overall, the QCNN can produce accurate multiclass classifications that are qualitatively similar to the classical model if the number of parameters is comparable. A similar level of accuracy has been achieved previously [31] for the 3-class classification problem. Here, we have demonstrated this level of the accuracy for the 4-class classification tasks, which to the best of our knowledge is the first such demonstration.

## 5 Conclusion

Here we have demonstrated the quantum multiclass classifier, which is based on the QCNN architecture. The main conceptual improvements that we have realized are the new model for quantum perceptron and an optimized structure of the quantum circuit. We have shown the use of the proposed approach for 4-class classification for the case of four MNISTs. As we have presented, the results obtained with the QCNN are comparable with those of CCNN for the case if the number of parameters are comparable. We expect that further optimizations of the perceptron can be studied in the future in order to make this approach more efficient. Moreover, since the scheme requires the use of multiqubit gates, the qudit processors, where multiqubit gate decompositions can be implemented in a more efficient manner, can be of interest for the realization of such algorithms.

## Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://deepai.org/dataset/mnist.

## Author contributions

DB—formulation of algorithm, AM—software development, AB—software development, DT—project management, AF—project supervision.

## Acknowledgments

We thank A. Gircha for useful comments. AM and AF acknowledge the support of the Russian Science Foundation (19-71-10092). This work was also supported by the Priority 2030 program at the National University of Science and Technology “MISIS” under the project K1-2022-027 (analysis of the method) and by the Russian Roadmap on Quantum Computing (testing the MNIST images).

## Conflict of interest

Authors DB, AM, AB, and AF were employed at the Russian Quantum Center which provides consulting services. Owing to the employments and consulting activities of authors, the authors have financial interests in the commercial applications of quantum computing.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

1. Fedorov AK, Gisin N, Beloussov SM, Lvovsky AI. Quantum computing at the quantum advantage threshold: A down-to-business review (2022), arXiv: 2203.17181v1 [quant-ph]. doi:10.48550/arXiv.2203.17181

2. Ladd TD, Jelezko F, Laflamme R, Nakamura Y, Monroe C, O'Brien JL. Quantum computers. *Nature* (2010) 464:45–53. doi:10.1038/nature08812

3. Brassard G, Chuang I, Lloyd S, Monroe C. Quantum computing. *Proc Natl Acad Sci U S A* (1998) 95:11032–3. doi:10.1073/pnas.95.19.11032

4. Shor P. In: Proceedings 35th Annual Symposium on Foundations of Computer Science (1994). p. 124–34.

5. Lloyd S. Universal quantum simulators. *Science* (1996) 273:1073–8. doi:10.1126/science.273.5278.1073

6. McArdle S, Endo S, Aspuru-Guzik A, Benjamin SC, Yuan X. Quantum computational chemistry. *Rev Mod Phys* (2020) 92:015003. doi:10.1103/revmodphys.92.015003

7. Gidney C, Ekerå M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* (2021) 5:433. doi:10.22331/q-2021-04-15-433

8. Preskill J. Quantum computing in the NISQ era and beyond. *Quantum* (2018) 2:79. doi:10.22331/q-2018-08-06-79

9. Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. *Phys Rev Lett* (2014) 113:130503. doi:10.1103/physrevlett.113.130503

10. Dunjko V, Briegel HJ. Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Rep Prog Phys* (2018) 81:074001. doi:10.1088/1361-6633/aab406

11. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. *Nature* (2017) 549:195–202. doi:10.1038/nature23474

12. Mottonen M, Vartiainen JJ, Bergholm V, Salomaa MM. Transformation of quantum states using uniformly controlled rotations. *Quan Inf Comput* (2005) 5: 467–73. doi:10.26421/qic5.6-5

13. Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, et al. Variational quantum algorithms. *Nat Rev Phys* (2021) 3:625–44. doi:10.1038/s42254-021-00348-9

14. Bharti K, Cervera-Lierta A, Kyaw TH, Haug T, Alperin-Lea S, Anand A, et al. Noisy intermediate-scale quantum algorithms. *Rev Mod Phys* (2022) 94:015004. doi:10.1103/revmodphys.94.015004

15. Lloyd S, Mohseni M, Rebentrost P. Quantum algorithms for supervised and unsupervised machine learning. (2013), arXiv:1307.0411v2 [quant-ph]. doi:10.48550/arXiv.1307.0411

16. Dunjko V, Taylor JM, Briegel HJ. Quantum-Enhanced machine learning. *Phys Rev Lett* (2016) 117:130501. doi:10.1103/physrevlett.117.130501

17. Amin MH, Andriyash E, Rolfe J, Kulchytskyy B, Melko R. Quantum Boltzmann machine. *Phys Rev X* (2018) 8:021050. doi:10.1103/physrevx.8.021050

18. Cong I, Choi S, Lukin MD. Quantum convolutional neural networks. *Nat Phys* (2019) 15:1273–8. doi:10.1038/s41567-019-0648-8

19. Zoufal C, Lucchi A, Woerner S. Quantum Generative Adversarial Networks for learning and loading random distributions. *Npj Quan Inf* (2019) 5:103. doi:10.1038/s41534-019-0223-2

20. Abbas A, Sutter D, Zoufal C, Lucchi A, Figalli A, Woerner S. The power of quantum neural networks. *Nat Comput Sci* (2021) 1:403–9. doi:10.1038/s43588-021-00084-1

21. Schuld M, Killoran N. Quantum machine learning in feature hilbert spaces. *Phys Rev Lett* (2019) 122:040504. doi:10.1103/physrevlett.122.040504

22. Schuld M, Bergholm V, Gogolin C, Izaac J, Killoran N. Evaluating analytic gradients on quantum hardware. *Phys Rev A (Coll Park)* (2019) 99:032331. doi:10.1103/physreva.99.032331

23. Mengoni R, Di Pierro A. Kernel methods in quantum machine learning. *Quan Mach Intell* (2019) 1:65–71. doi:10.1007/s42484-019-00007-4

24. Schuld M, Fingerhuth M, Petruccione F. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Letters)* (2017) 119: 60002. doi:10.1209/0295-5075/119/60002

25. Benedetti M, Realpe-Gómez J, Perdomo-Ortiz A. Quantum-assisted helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices. *Quan Sci Technol* (2018) 3:034007. doi:10.1088/2058-9565/aabd98

26. Grant E, Benedetti M, Cao S, Hallam A, Lockhart J, Stojevic V, et al. Hierarchical quantum classifiers. *Npj Quan Inf* (2018) 4:65. doi:10.1038/s41534-018-0116-9

27. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. *Nature* (2019) 567: 209–12. doi:10.1038/s41586-019-0980-2

28. Tacchino F, Macchiavello C, Gerace D, Bajoni D. An artificial neuron implemented on an actual quantum processor. *Npj Quan Inf* (2019) 5:26. doi:10.1038/s41534-019-0140-4

29. Johri S, Debnath S, Mocherla A, Singh A, Prakash A, Kim J, et al. Nearest centroid classification on a trapped ion quantum computer (2020), arXiv: 2012.04145 [quant-ph]. doi:10.48550/arXiv.2012.04145

30. Li W, Deng D-L. Recent advances for quantum classifiers. *Sci China Phys Mech Astron* (2021) 65:220301. doi:10.1007/s11433-021-1793-6

31. Chalumuri A, Kune R, Manoj BS. A hybrid classical-quantum approach for multi-class classification. *Quan Inf Process* (2021) 20:119. doi:10.1007/s11128-021-03029-9

32. Hur T, Kim L, Park DK. Quantum convolutional neural network for classical data classification. *Quan Mach Intell* (2022) 4:3. doi:10.1007/s42484-021-00061-x

33. Sels D, Dashti H, Mora S, Demler O, Demler E. Quantum approximate Bayesian computation for NMR model inference. *Nat Mach Intell* (2020) 2: 396–402. doi:10.1038/s42256-020-0198-x

34. Seetharam K, Biswas D, Noel C, Risinger A, Zhu D, Katz O, et al. Digital quantum simulation of nmr experiments (2021). arXiv:2109.13298 [quant-ph]. doi:10.48550/arXiv.2109.13298

35. Jain S, Ziauddin J, Leonchyk P, Yenkanchi S, Geraci J. Quantum and classical machine learning for the classification of non-small-cell lung cancer patients. *SN Appl Sci* (2020) 2:1088. doi:10.1007/s42452-020-2847-4

36. Albash T, Lidar DA. Demonstration of a scaling advantage for a quantum annealer over simulated annealing. *Phys Rev X* (2018) 8:031016. doi:10.1103/physrevx.8.031016

37. Enos GR, Reagor MJ, Henderson MP, Young C, Horton K, Birch M, et al. Synthetic weather radar using hybrid quantum-classical machine learning (2021). arXiv:2111.15605 [quant-ph]. doi:10.48550/arXiv.2111.15605

38. Perdomo-Ortiz A, Benedetti M, Realpe-Gómez J, Biswas R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quan Sci Technol* (2018) 3:030502. doi:10.1088/2058-9565/aab859

39. Broughton M, Verdon G, McCourt T, Martinez AJ, Yoo JH, Isakov SV, et al. Tensorflow quantum: A software framework for quantum machine learning (2020), arXiv:2003.02989v2 [quant-ph]. doi:10.48550/arXiv.2003.02989

40. Möttönen M, Vartiainen JJ, Bergholm V, Salomaa MM. Quantum circuits for general multiqubit gates. *Phys Rev Lett* (2004) 93:130502. doi:10.1103/physrevlett.93.130502

41. Kiktenko EO, Nikolaeva AS, Xu P, Shlyapnikov GV, Fedorov AK. Scalable quantum computing with qudits on a graph. *Phys Rev A (Coll Park)* (2020) 101: 022304. doi:10.1103/physreva.101.022304

42. Liu W-Q, Wei H-R, Kwek L-C. Implementation of cnot and toffoli gates with higher - dimensional spaces. (2021), arXiv:2105.10631v3 [quant-ph].

43. Nikolaeva AS, Kiktenko EO, Fedorov AK. Efficient realization of quantum algorithms with qudits (2021), arXiv:2111.04384v2 [quant-ph]. doi:10.48550/arXiv.2111.04384

44. Nikolaeva AS, Kiktenko EO, Fedorov AK. Decomposing the generalized toffoli gate with qutrits. *Phys. Rev. A* (2022) 105:032621. doi:10.1103/PhysRevA.105.032621

45. Gokhale P, Baker JM, Duckering C, Brown NC, Brown KR, Chong FT. In: Proceedings of the 46th International Symposium on Computer Architecture, ISCA '19. New York, NY, USA: Association for Computing Machinery (2019). p. 554–66.

46. Huang HY, Broughton M, Mohseni M, Babbush R, Boxio S, Neven H, et al. Power of data in quantum machine learning. *Nat Commun* (2021) 12:2631. doi:10.1038/s41467-021-22539-9