Quantum machine learning for image classification

Arsenii Senokosov, Alexander Sedykh, Asel Sagingalieva, and Alexey Melnikov

Terra Quantum AG, Kornhausstrasse 25, 9000 St. Gallen, Switzerland

Image recognition and classification are fundamental tasks with diverse practical applications across various industries, making them critical in the modern world. Recently, machine learning models, particularly neural networks, have emerged as powerful tools for solving these problems. However, the utilization of quantum effects through hybrid quantum-classical approaches can further enhance the capabilities of traditional classical models. Here, we propose two hybrid quantum-classical models: a neural network with parallel quantum layers and a neural network with a quanvolutional layer, which address image classification problems. One of our hybrid quantum approaches demonstrates remarkable accuracy of more than 99% on the MNIST dataset. Notably, in the proposed quantum circuits all variational parameters are trainable, and we divide the quantum part into multiple parallel variational quantum circuits for efficient neural network learning. In summary, our study contributes to the ongoing research on improving image recognition and classification using quantum machine learning techniques. Our results provide promising evidence for the potential of hybrid quantum-classical models to further advance these tasks in various fields, including healthcare, security, and marketing.

Introduction

Image classification is a critical task in the modern world due to its wide range of practical applications in various fields [1]. For instance, in medical imaging, image classification algorithms have been shown to significantly improve the accuracy and speed of diagnoses of many diseases [2, 3]. In the field of autonomous vehicles, image classification plays a crucial role in object detection, tracking, and classification, which is necessary for safe and efficient navigation.

Deep learning approaches [4] like deep convolutional neural networks (CNNs) have emerged as powerful tools for image classification and recognition tasks [5, 6], achieving state-of-the-art performance on various benchmark datasets [7, 8]. However, as the amount of visual data increases, modern neural networks are facing significant computational challenges.

Quantum technologies, on the other hand, offer the potential to overcome this computational limitation by harnessing the power of quantum mechanics to perform computations in parallel [9]. Quantum machine learning (QML) is a rapidly evolving field that combines the principles of quantum mechanics and classical machine learning [10, 11]. This field has the potential to revolutionize various areas of computing, including image classification [12, 13]. It has attracted significant attention due to its potential to solve computational problems that classical computers are unable to solve efficiently [9]. This potential arises from the unique features of quantum computing, such as superposition and entanglement, which can provide an exponential speedup for specific machine learning tasks [14]. Moreover, QML algorithms produce probabilistic results, which is very natural for classification problems [15] and also act in an exponentially bigger search space, which greatly increases their performance [16–18]. However, the real-world implementation of quantum algorithms faces significant challenges, such as the need for error correction and the high sensi-

tivity of quantum systems to external disturbances [19]. Despite these challenges, QML has shown promising results in several applications [20]. In the context of image classification, QML algorithms can process large datasets of images more efficiently than classical algorithms, leading to faster and more accurate classification [21].

A promising area of research within QML for image classification is the hybrid quantum neural network (HQNN) [14]. HQNNs combine classical deep learning architectures with QML algorithms [22–26], namely Variational Quantum Circuits (VQCs), creating a hybrid system that leverages the strengths of both classical and quantum computing. This approach allows for the processing of large datasets with greater efficiency than classical deep learning architectures alone [27]. HQNNs have shown promise in a variety of tasks including image classification [28], regression problems [29], even satellite mission planning [30] and personalized medicine [31]. Further research is needed to explore the full potential of HQNNs in image classification and to develop more robust and scalable algorithms.

In this article, we propose two approaches to leverage quantum computing in the field of image recognition. The first approach involves applying parallel VQCs after classical deep convolutional layers, while the second approach involves using a HQNN with a quanvolutional layer. We evaluate the performance of these hybrid models on the MNIST dataset of hand-written digits, which is described in Section A, and demonstrate their ability to classify images.

The first model (described in Section B) combines classical convolutional layers with parallel quantum layers (HQNN-Parallel). The quantum part is analogous to a classical fully connected layer. We compare the hybrid model with its most closely corresponding classical counterpart (in terms of the architecture and the number of layers) and observe that the hybrid model outperforms the classical model in accuracy (achieving 99.21% accuracy) despite having eight times fewer parameters.

In the second model (described in Section C), we introduce HQNN with quanvolutional layer (HQNN-Quanv), which is a filter that applies a convolution to the input image and reduces its resolution. The HQNN-Quanv achieves a similar accuracy to the classical model (67% accuracy) despite having four times fewer trainable parameters in the first layer compared to the classical counterpart. Additionally, the hybrid model outperforms the classical model with the same number of weights.

Every parameter in both of our models is trainable, which allows us to achieve such remarkable accuracy. This highlights the potential of quantum computing and quantum machine learning (QML) in advancing the field of image recognition. Our results contribute to the ongoing research in this area and demonstrate the exciting possibilities for the future of QML in other fields.

Results

A. Dataset

This section describes the dataset we use which is called Modified National Institute of Standards and Technology (MNIST) [32]. The MNIST database consists of a large collection of gray-scale handwritten numbers, ranging from 0 to 9. Sample images from the dataset are presented in Fig. 1. Each image has a resolution of 28×28 pixels, and the main objective is to classify each image by assigning a class label using a neural network. In other words, the task is to recognize which digit is present in the image. This dataset is widely used for making first steps in the sphere of machine learning. Nevertheless, it is worth-studying as it helps test the performance of various neural networks models [33, 34], especially models with VQCs. The MNIST dataset used in this study comprises a total of 70000 images, with 60000 images reserved for training and 10000 images for testing. However, in certain cases, it may be advantageous to reduce the number of images in order to expedite the training process and gain immediate insights into the model's performance.

Despite being a widely used dataset, the MNIST database contains a few images that are broken or ambiguous, making it challenging for even humans to make a clear judgment. Fig. 2 provides examples of such images. However, our introduced hybrid model can determine what the number is in the image with over 99% accuracy.

B. Hybrid Quantum Neural Network with parallel quantum dense layers, HQNN-Parallel

This section describes our first proposed model, the Hybrid Quantum Neural Network with parallel quantum dense layers, each of which is a VQC. Section B 3 shows the results and comparison of the hybrid model with its
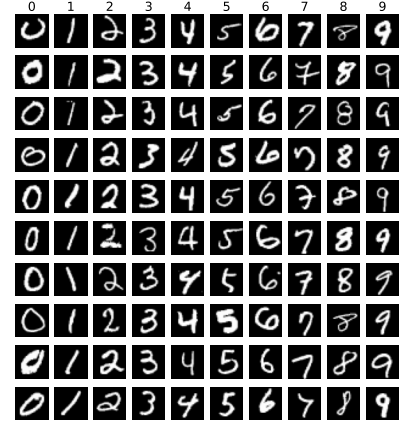

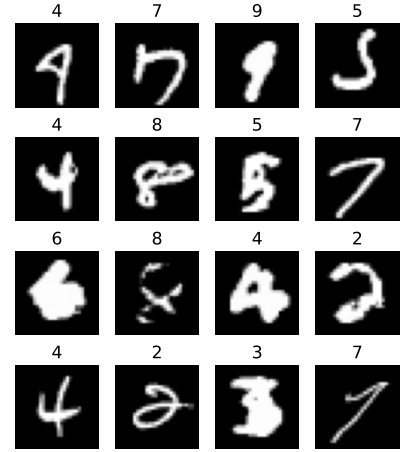FIG. 1: Examples of images from the MNIST dataset

FIG. 2: Examples of ambiguous images from the MNIST dataset.

classical counterpart, CNN B 4. The HQNN-Parallel consists of two main components: a classical convolutional block B 1 and a combination of classical fully connected and parallel quantum layers B 2. The primary objective of the classical convolutional block is to reduce the dimensionality of the input data and prepare it for subsequent processing. The classical fully connected and parallel quantum layers constitute the core of the HQNN-Parallel, and are responsible for prediction tasks of the model. Further details on the architecture and implementation of the HQNN-Parallel are presented in subsequent sections.

1. Classical Convolutional Layers

Fig. 3 depicts the general structure of the classical convolutional part of the proposed HQNN-Parallel. The convolutional part of the network is comprised of two main blocks, followed by fully-connected layers. In this study, we utilized Rectified Linear Unit (ReLU) as the activa-
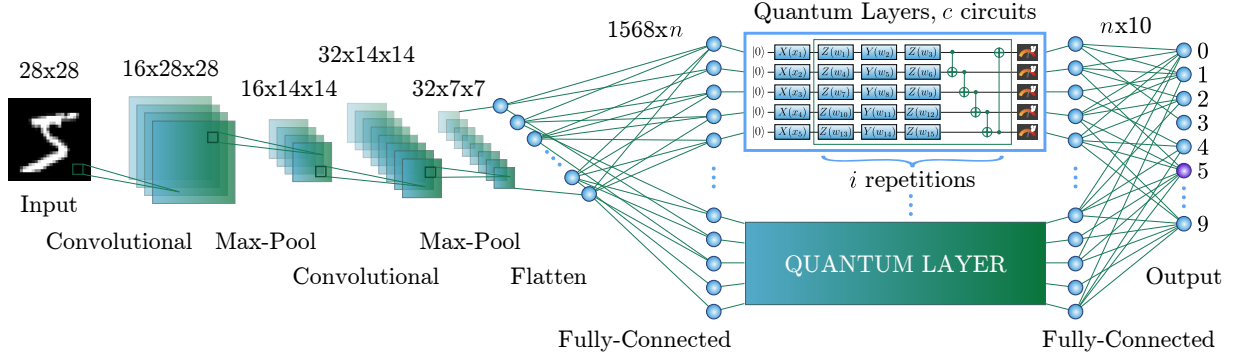
FIG. 3: Architecture of the proposed HQNN-Parallel. The input data samples are transformed by a series of convolutional layers, which extract relevant features from the input and reduce the input dimensionality. The output channels of the convolutional layers are then flattened into a single vector and fed into the dense part of the HQNN-Parallel. The hybrid dense part contains both classical and quantum layers. The quantum layers are implemented using parallel VQCs, which allow for simultaneous execution, reducing the total computation time. The output of the last classical fully connected layer is a predicted digit between 0 and 9.

tion function [35]. Batch Normalization [36] is employed in the network as it stabilizes the training process and improves the accuracy of the model.

The first block of the convolutional part of the HQNN-Parallel comprises a convolutional layer with one input channel and 16 output channels, utilizing a square kernel of size 5×5. The layer operates with a stride of one pixel and applies a two-pixel padding to the input data. Batch Normalization is applied to the output of the convolutional layer, followed by an activation function (ReLU) and MaxPooling [37] with a kernel size of two pixels. The resulting feature map has dimensions of $16 \times 14 \times 14$ pixels.

The second block contains a convolutional layer with 16 input channels and 32 output channels, utilizing the same kernel size and padding as the previous layer. The MaxPooling parameters remain unchanged, resulting in a feature map with dimensions of $32 \times 7 \times 7$ pixels, which will become an input for the fully connected part of the network.

2. Hybrid Dense Layers

Following the convolutional part, the HQNN- Parallel, continues with a hybrid dense part, as shown in Fig. 3. The $32 \times 7 \times 7$ feature map produced by the convolutional part serves as input for the first dense layer, which transforms the feature map from 1568 to m features. The value of m is determined by the chosen quantum part and represents the total number of encoding parameters in the quantum layers.

Each quantum layer is designed to maintain the number of input and output features, and the output of the quantum layer is fed into the second classical fully connected layer. This layer performs the final transformation and maps the m input features to 10 output features, corresponding to the number of classes into which the

images can be classified. After each classical dense layer, Batch Normalization and ReLU activation are applied.

It is worth noting that the structure of the HQNN-Parallel, including the number of layers and the number of features, can be adjusted to optimize the performance on a specific task.

3. Structure of Quantum Layer

The quantum component of the proposed HQNN-Parallel, depicted in the Fig. 3, consists of c parallel quantum layers, each of which is a VQC composed of three parts: embedding, variational gates, and measurement. The input data to the quantum layers are m features from the previous classical fully connected layer, divided into c parts, with each part being a vector of q values, $x = (\phi_1, \phi_2, ..., \phi_q) \in \mathbb{R}^q$. To encode these classical features into quantum Hilbert space, we use the "angle embedding" method, which rotates each qubit in the ground state around the X-axis on the Bloch sphere [38] by an angle proportional to the corresponding value in the input vector: $|\psi\rangle = R_x^{emb}(x) |\psi_0\rangle$, where $|\psi_0\rangle = |0\rangle^{\otimes q}$. This operation encodes the input vector into quantum space, and the resulting quantum state represents the input data from the previous classical layer. It is important to note that m is divisible by q, since the input data vector is divided into $c = m/q$ parts, with each part serving as input to a VQC.

The encoding part for each VQC is followed by a variational part, which consists of two parts: rotations with trainable parameters and subsequent CNOT operations [39]. The rotations serve as quantum gates that transform the encoded input data according to the variational parameters, while the CNOT operations entangle the qubits in the VQC. The depth of the variational part, denoted as i, is a hyperparameter that determines the number of iterations of the rotations and CNOT op-
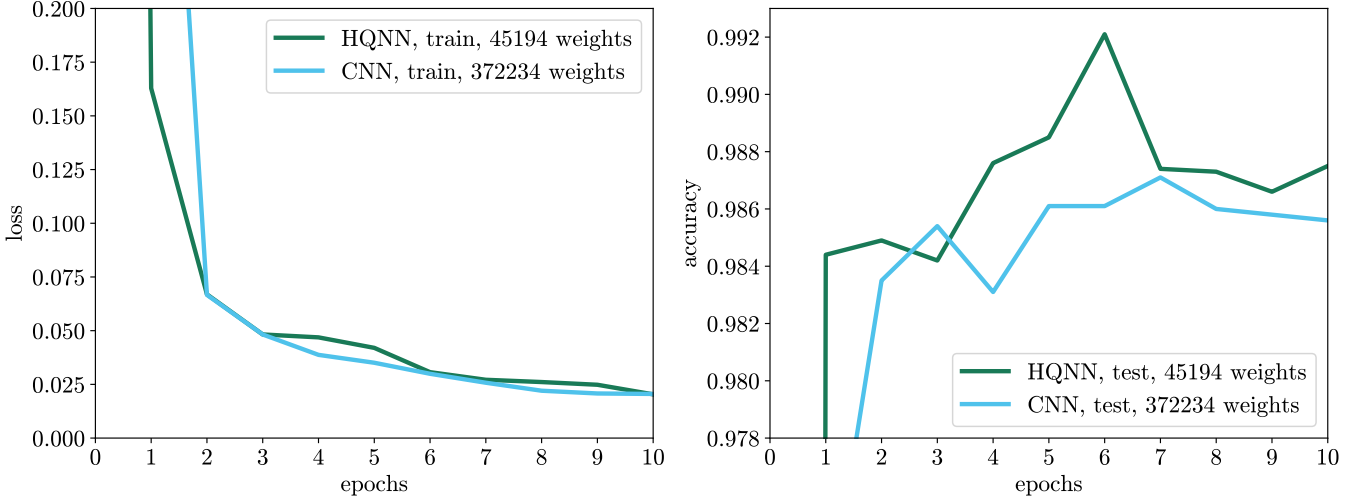
FIG. 4: Train and test results for the HQNN-Parallel and the CNN. The HQNN has a 99.21% accuracy on the test data and outperforms the CNN which has a 98.71% accuracy even though the classical model has 8 times more variational parameters than the hybrid one.

erations in the VQC. It is important to note that the variational parameters for each VQC are different in each of the i repetitions and for each of c quantum circuits. Thus, the total number of weights in the quantum part of the HQNN-Parallel is calculated as $q \cdot 3i \cdot c$.

After performing these operations, measurement in the Pauli basis matrices is performed, resulting in

$$v^{(j)} = \langle 0| \, R_x^{emb}(\phi_j)^\dagger U(\theta)^\dagger Y_j U(\theta) R_x^{emb}(\phi_j) \, |0\rangle, \quad (1)$$

where Y_j is the Pauli-Y matrix for the jth qubit, $R_x^{emb}(\phi_j)$ and $U(\theta)$ are operations, performed by the embedding and trainable parts of the VQC, respectively, and θ is a vector of trainable parameters. After this operation, we have the vector $v \in \mathbb{R}^q$. The outputs of all the VQCs would be concatenated to form new vector $\hat{v} \in \mathbb{R}^m$ that is the input data for a subsequent classical fully-connected layer. This layer, being the final layer in the classification pipeline, produces an output in the form of probability distribution over the set of classes. In our case, each input image is associated with one of the ten possible digits from 0 to 9, and the output of each neuron represents the probability that the image belongs to that class. The neuron with the highest output probability is selected as the predicted class for the image.

4. Training and results

Model	train loss	test loss	test acc	param num
CNN	0.0205	0.0449	98.71	372234
HQNN	0.0204	0.0274	99.21	45194

TABLE I: Summary of the results for the HQNN-Parallel and its classical analogue, CNN.
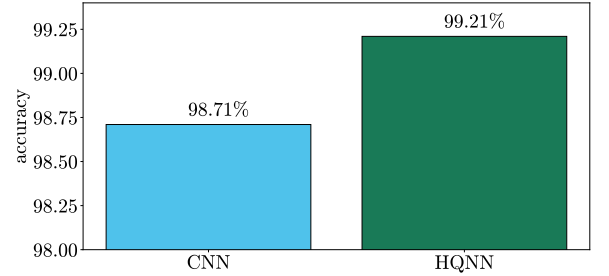

FIG. 5: Test accuracies for the HQNN-Parallel and its classical analogue CNN.

As described above, the HQNN-Parallel will be trained on MNIST dataset A. No preprocessing is applied, so the entire collection is used for training (60000 images are in the training set and 10000 are in the test set). In the context of training the proposed HQNN-Parallel, the ultimate objective is to minimize the loss function during the optimization process. The cross-entropy function is employed as the loss function, given by:

$$l = -\sum_{c=1}^{k} y_c \log p_c, \quad (2)$$

where p_c is the prediction probability, y_c is either 0 or 1, determining respectively if the image belongs to the prediction class, and k is the number of classes.

The parameters of the classical layers are optimized using the backpropagation algorithm [40], which is automatically implemented in the PyTorch library [41]. The backpropagation algorithm is used to calculate the gradients of the loss function with respect to the parameters of the network, allowing for their optimization via gradient descent. However, the use of quantum layers in this task
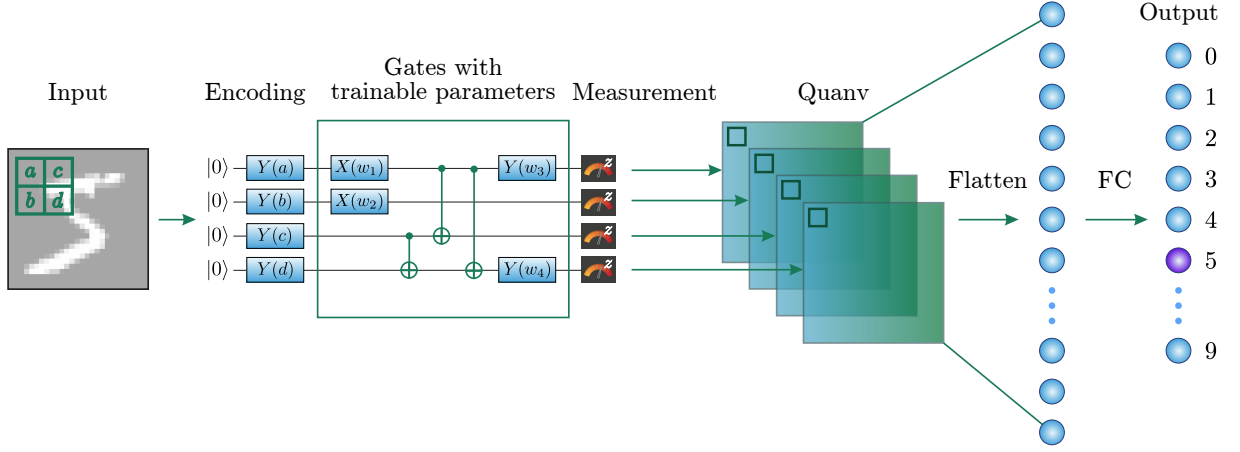
FIG. 6: Architecture of the HQNN-Quanv. The quanvolutional kernel encodes a chunk of pixels in the input image into the quantum circuit via angle embedding with R_y gates. Then some rotation gates with trainable parameters and CNOTs are applied to the qubits, carrying image data. Measuring each wire with σ_z expectation value yields an output pixel. As we have 4 qubits, the amount of output channels is also 4. All of output channels are flattened into one vector and fed into the fully connected layer, which gives a predicted digit from 0 to 9.

is more complex than classical methods for computing gradients. To overcome this challenge, we employ the PennyLane framework [42], which provides access to a variety of optimization techniques. We utilize the parameter shift rule [43], which is compatible with physical implementations of quantum computing [44]. This method involves evaluating the gradient of a quantum circuit by shifting the parameters in the circuit and computing the corresponding change in the circuit's output. The resulting gradient can then be used to update the circuit's parameters and iteratively minimize the loss function. By using the parameter shift rule, we are able to efficiently optimize the variational parameters in the quantum layers of the HQNN, enabling the network to learn complex patterns in the input data and achieve accurate results.

In the process of solving the problem, we tried various architectures of quantum layers. The most successful architecture for the HQNN-Parallel used a quantum layer with 5 qubits and 3 repetitions of the strongly entangling layers. The number of quantum layers equals 4.

The HQNN-Parallel managed to achieve a 99.21% accuracy. In order to compare the performance of the HQNN with a classical CNN, the convolutional part of the HQNN was held constant, while the quantum part was replaced with a classical dense layer containing m neurons. This modified CNN was then trained on the same MNIST dataset. A comparison of the training outcomes is depicted in Fig. 4.

The trainable parameters, as well as the primary training and testing results, for both the HQNN-Parallel and the CNN are summarized in Table I and illustrated in Fig. 5. From these results, it is evident that the most successful implementation of the HQNN-Parallel surpasses the performance of a CNN that possesses approximately 8 times more parameters.

C. Hybrid Quantum Neural Network with quanvolutional layer, HQNN-Quanv

In this section, we give a detailed description of our second hybrid quantum approach for solving the problem of recognizing numbers from the MNIST dataset, based on the combination of a quanvolutional layer and classical fully connected layers. The scheme of this network is presented in Fig. 6. Also, we compare our hybrid model with its classical analogue CNN, investigate the relationship between quanvolutional and convolutional layers as well as their dependence on the number of output channels.

1. Quanvolutional layer

The general architecture of a quanvolutional layer [45] is shown in Fig. 6. Similar to classical convolutional layers, the quanvolutional layer comprises a filter of size $n \times n$ pixels that convolves the input image, producing a lower-resolution output image. However, the quanvolutional layer is unique in a sense that its filter is implemented using a quantum circuit consisting of n qubits. The circuit can be decomposed into three distinct parts: classical-to-quantum data encoding, variational gates, and a quantum measurement. These parts work together to determine the filter's action on the input image.

There are plenty of encoding (embedding) methods to transfer classical data into quantum states. In this section, as in the previous one, we use the "angle embedding" technique. It is achieved by rotating the qubits from their initial $|0\rangle$ value with the $R_y(\varphi)$ unitaries, where φ is determined by the value of the corresponding pixel. After the classical data is encoded, the quantum states undergo unitary transformations, defined by the
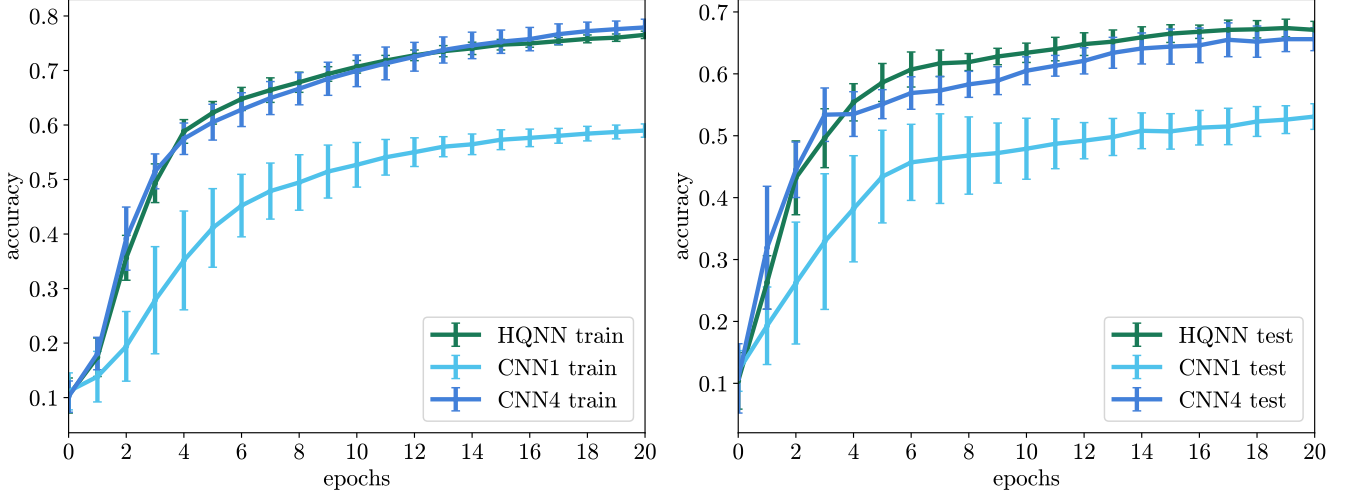
FIG. 7: Train and test accuracies for the CNN and HQNN-Quanv models with stride set to 4. The models differ only in the kernel and the number of output channels. (a) HQNN: Quanvolutional kernel with 1 input channel, 4 output channels; (b) CNN1: Convolutional kernel with 1 input channel, 1 output channel; (c) CNN4: Convolutional kernel with 1 input channel, 4 output channels. The HQNN-Quanv with an accuracy of $67 \pm 1\%$ on test data outperforms the CNN1 which has an accuracy of $53 \pm 2\%$ and the CNN4 which has an accuracy of $66 \pm 2\%$, although the CNN1 has the same number of weights in the filter as in the hybrid model and CNN4 has 4 times more weights than in the hybrid model.

variational part.

The variational part in the quanvolutional layer usually consists of arbitrary single-qubit rotations and CNOT gates, arranged in a particular way determined by the researcher. The unitaries in the VQC are parameterized by a set of variational parameters, which are learned via training the neural network. The ultimate goal of the model training is to find a measurement basis (via tweaking variational gate parameters) that tells us the most information about a fragment of a picture confined by the quantum filter.

Finally, for each wire, the expectation value of an arbitrary operator is calculated to obtain the classical output. As it is a real number, it represents the filter's output pixel, while each wire yields a different image channel. For instance, a quanvolutional filter of size 2×2 has 4-qubit circuit, which transforms 1 input image into 4 images of reduced size.

2. Structure of HQNN-Quanv

This subsection details the architecture of the HQNN-Quanv, which is shown in Fig. 6. At first, a simple angle embedding of the classical data via $R_y(\varphi)$ single-qubit rotations on each wire is used, where the original pixel value $[0, 1]$ is scaled to $\varphi \in [0, \pi]$. Then, we have a variational circuit part, which consists of 4 single-qubit rotations, parameterized with trainable weights, as well as three CNOT gates. At the end of the circuit, we measure the expectation value $\langle \sigma_z \rangle$ of the Pauli-Z operator on each qubit. Each channel is a picture with 4×4 pixels. After that, four output channels are flattened and fed into fully

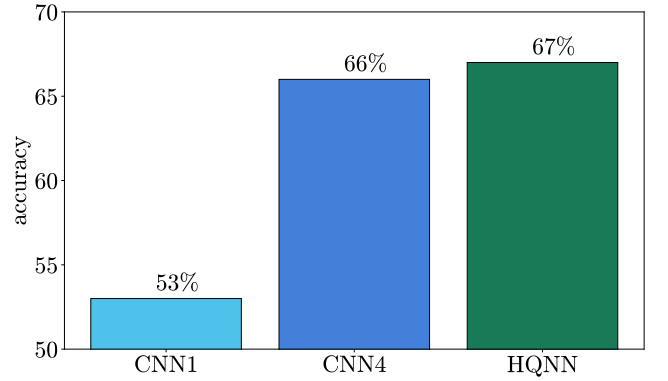connected layer, which yields a digit's probability.

3. Training and results

FIG. 8: Test accuracies for HQNN-Quanv ($67 \pm 1\%$), CNN1 ($53 \pm 2\%$) and CNN4 ($66 \pm 2\%$). The HQNN outperforms the CNN1, which has the same number of variational parameters. The HQNN's accuracy score is equivalent to CNN4's, which has 4 times many weights in its filter.

In this section, we describe the training process. In order to reduce the training time of the HQNN-Quanv, only 600 images from the MNIST dataset A are used with 500 of them acting as training data and 100 as test data. We also use PyTorch's resize transform with bilinear interpolation to downscale images from 28×28 to 14×14 pixels. We still use a cross-entropy loss function.

While the classical model has only one way of training weights via backpropagation, the HQNN has several options, such as the parameter-shift rule, adjoint differentiation [46] or backpropagation (which, of course, is impossible on a real quantum computer). Adjoint differentiation seems to have the most favorable scaling with both layers and wires [47], but on this particular circuit (Fig. 6) backpropagation proved to be quicker.

Considering everything stated above, let us see the results of the training. We trained two CNN's with different numbers of output channels and one HQNN for 20 epochs (Fig. 7). The models were intentionally made simple and had sufficiently few parameters so as to avoid overfitting on the relatively small dataset. The test accuracy of these models is presented in Fig. 8. For each epoch, the accuracy is averaged over 10 models with random initial weights. The error bars depict one standard deviation. Hence, for each epoch we have a mean accuracy over 10 equivalent models and an error bar for the standard deviation.

At the end of the training, the HQNN-Quanv had a test accuracy of 0.67 ± 0.01, which is close enough to the CNN4 result of 0.66 ± 0.02, while CNN1 had 0.53 ± 0.02. The HQNN model has only 4 trainable weights in its quanvolutional kernel, which parameterize rotation gates in the VQC. CNN1 and CNN4 have 4 and 16 trainable parameters in their convolutional kernels, respectively. Therefore, the HQNN's performance based on the accuracy score is equivalent to CNN4's, which has 4 times many weights in its filter.

Discussion

In this work, we introduced two hybrid approaches to image classification. The first approach was a HQNN-Parallel. This method allowed us to classify handwritten images of digits from the MNIST dataset with an accuracy of more than 99%. It should be noted that the number of weights in the quantum model was 8 times less than in its classical counterpart, which achieved an accuracy of only 98.71%. Also the successful implementation of parallel variational quantum circuits in the hybrid model was demonstrated, which led to such remarkable results. Our proposed architecture is a unique combination of classical and quantum layers, which we believe to be a breakthrough in solving image classification problems.

The second approach we presented was a HQNN-Quanv. The quanvolutional layer needs significantly fewer weights, 4 times less than the classical analogue, to achieve approximately the same classification accuracy ($67 \pm 1\%$ for the hybrid model versus $66 \pm 2\%$ for the classical one on the test samples when averaged over 10 models), while the classical analogue with the same number of variational parameters as the hybrid model achieves an accuracy of $53 \pm 2\%$.

Further research is needed to explore the full potential of HQNNs for image classification, including testing on larger datasets and more complex architectures. Additionally, the development of more efficient optimization techniques for training VQCs and the implementation of larger-scale quantum hardware could lead to even more significant performance improvements.

In summary, our developments provide two hybrid approaches to image classification that demonstrate the power of combining classical and quantum methods. Our proposed models show improved performance over classical models with similar architectures while using significantly fewer parameters. We believe that these results pave the way for further research in developing hybrid models that utilize the strengths of both classical and quantum computing.

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv preprint arXiv:1506.01497, 2015.

[2] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. Medical Image Analysis, 42:60–88, 2017.

[3] Lei Tian, Brady Hunt, Muyinatu A. Lediju Bell, Ji Yi, Jason T. Smith, Marien Ochoa, Xavier Intes, and Nicholas J. Durr. Deep Learning in Biomedical Optics. Lasers in Surgery and Medicine, 53(6):748–775, 2021.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv preprint arXiv:1511.00561, 2015.

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. arXiv preprint arXiv:1608.06993, 2016.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[8] Joseph Redmon and Ali Farhadi. YOLO9000: Better, Faster, Stronger. arXiv preprint arXiv:1612.08242, 2016.

[9] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien. Quantum computers. Nature, 464(7285):45–53, 2010.

[10] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics, 81(7):074001, 00 2018.

[11] Alexey Melnikov, Mohammad Kordzanganeh, Alexander Alodjants, and Ray-Kuang Lee. Quantum machine learning: from physics to software engineering. Advances in Physics: X, 8(1):2165452, 2023.

[12] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. Contemporary Physics, 56(2):172–185, 2015.

[13] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. Nature, 549(7671):195–202, 2017.

[14] Maria Schuld, Mark Fingerhuth, and Francesco Petruccione. Implementing a distance-based classifier with a quantum interference circuit. arXiv preprint arXiv:1703.10793, 2017.

[15] Denis Bokhan, Alena S. Mastiukova, Aleksey S. Boev, Dmitrii N. Trubnikov, and Aleksey K. Fedorov. Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. Frontiers in Physics, 10:1069985, 2022.

[16] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411, 2013.

[17] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science, pages 124–134, 1994.

[18] Seth Lloyd. Universal quantum simulators. Science, 273:1073–1078, 1996.

[19] Scott Aaronson and Lijie Chen. Complexity-Theoretic Foundations of Quantum Supremacy Experiments. arXiv preprint arXiv:1612.05903, 2016.

[20] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum Deep Learning. arXiv preprint arXiv:1412.3489, 2014.

[21] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. Nature, 567(7747):209–212, 2019.

[22] Simon C Marshall, Casper Gyurik, and Vedran Dunjko. High dimensional quantum learning with small quantum computers. arXiv preprint arXiv:2203.13739, 2022.

[23] Mohammad Kordzanganeh, Pavel Sekatski, Leonid Fedichkin, and Alexey Melnikov. An exponentially-growing family of universal quantum circuits. arXiv preprint arXiv:2212.00736, 2022.

[24] Adrián Pérez-Salinas, Radoica Draškić, Jordi Tura, and Vedran Dunjko. Reduce-and-chop: Shallow circuits for deeper problems. arXiv preprint arXiv:2212.11862, 2022.

[25] Mohammad Kordzanganeh, Daria Kosichkina, and Alexey Melnikov. Parallel hybrid networks: an interplay between quantum and classical neural networks. arXiv preprint arXiv:2303.03227, 2023.

[26] Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. Nature Communications, 14(1):517, 2023.

[27] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum Circuit Learning. Physical Review A, 98(3):032309, 2018.

[28] Asel Sagingalieva, Andrii Kurkin, Artem Melnikov, Daniil Kuhmistrov, Michael Perelshtein, Alexey Melnikov, Andrea Skolik, and David Von Dollen. Hyperparameter optimization of hybrid quantum neural networks for car classification. arXiv preprint arXiv:2205.04878, 2022.

[29] Michael Perelshtein, Asel Sagingalieva, Karan Pinto, Vishal Shete, Alexey Pakhomchik, et al. Practical application-specific advantage through hybrid quantum computing. arXiv preprint arXiv:2205.04858, 2022.

[30] Serge Rainjonneau, Igor Tokarev, Sergei Iudin, Saaketh Rayaprolu, Karan Pinto, et al. Quantum algorithms applied to satellite mission planning for Earth observation. arXiv preprint arXiv:2302.07181, 2023.

[31] Asel Sagingalieva, Mohammad Kordzanganeh, Nurbolat Kenbayev, Daria Kosichkina, Tatiana Tomashuk, and Alexey Melnikov. Hybrid quantum neural network for drug response prediction. arXiv preprint arXiv:2211.05777, 2022.

[32] Li Deng. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6):141–142, 2012.

[33] Dan Claudiu Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. Neural Computation, 22(12):3207–3220, 2010.

[34] D Ciresan, U Meier, and J Schmidhuber. Multi-column Deep Neural Networks for Image Classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 1:3642–3649, 2012.

[35] Abien Fred Agarap. Deep learning using rectified linear units (ReLU). arXiv preprint arXiv:1803.08375, 2018.

[36] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[38] F. Bloch. Nuclear Induction. Physical Review, 70(7-8):460–474, 1946.

[39] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. Physical Review A, 52(5):3457–3467, 1995.

[40] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533–536, 10 1986. Number: 6088 Publisher: Nature Publishing Group.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, et al. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

[42] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968, 2018.

[43] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. Quantum, 6:677, 2022.

[44] Mohammad Kordzanganeh, Markus Buchberger, Maxim Povolotskii, Wilhelm Fischer, Andrii Kurkin, Wilfrid Somogyi, Asel Sagingalieva, Markus Pflitsch, and Alexey Melnikov. Benchmarking simulated and physical quantum processing units using quantum and hybrid algorithms. arXiv preprint arXiv:2211.15631, 2022.

[45] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. arXiv preprint arXiv:1904.04767, 2019.

[46] Tyson Jones and Julien Gacon. Efficient calculation of gradients in classical simulations of variational quantum algorithms. arXiv preprint arXiv:2009.02823, 2020.

[47] Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, and Lei Wang. Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design. Quantum, 4:341, October 2020.