Neural Networks with Variational Quantum Circuits

Syed Muhammad Abuzar Rizvi, Muhammad Shohibul Ulum, Naema Asif, and Hyundong Shin(✉)

Department of Electronics and Information Convergence Engineering, Kyung Hee University, Yongin 17104, Republic of Korea
hshin@khu.ac.kr

Abstract. The field of machine learning is an interdisciplinary area that aims to extract useful information from data through mathematical means. Integrating quantum computing with machine learning has led to exciting new avenues of research, where quantum mechanics principles are applied to enhance and optimize classical machine learning algorithms. In this study, we explore hybrid quantum-classical neural networks with an approach that combines both classical and quantum computing. We achieve this by implementing a variational quantum circuit as the output layer of a classical convolutional neural network. We use this hybrid neural network to classify images of digits from the MNIST dataset. Using this approach, we were able to classify images with high accuracy. Furthermore, due to its flexibility, this hybrid algorithm can be adapted to explore the potential of quantum computing especially in the era of noisy intermediate-scale quantum devices.

Keywords: Quantum Computing · Variational Quantum Circuits · Neural Networks · Image Classification

1 Introduction

Machine learning (ML) has become a fundamental aspect of modern computing, enabling computers to learn from data and make predictions or decisions based on that knowledge. ML algorithms have applications in various fields, including computer vision, natural language processing, speech recognition, recommendation systems, and quantum science, among others [11,23]. However, the growing complexity of modern data sets has made it increasingly challenging to develop

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2019R1A2C2007037, NRF-2022R1A4A3033401) and by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2021-0-02046) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This research is partially funded by the BK21 FOUR program of National Research Foundation of Korea.

N.-S. Vo and H.-A. Tran (Eds.): INISCOM 2023, LNICST 531, pp. 203–214, 2023.
https://doi.org/10.1007/978-3-031-47359-3_15

accurate and efficient ML models using classical computing techniques. Quantum machine learning (QML) is a rapidly emerging field that seeks to address some of the limitations of classical ML by harnessing the power of quantum computing. [6,7,25]. Quantum computing is a paradigm of computing that exploits the principles of quantum mechanics to perform certain calculations much faster than classical computers [8,17]. By combining the two fields, quantum-enhanced ML promises to develop more accurate and efficient ML models that can process larger data sets and solve more complex problems [10]. Moreover, quantum computers have the ability to extract intricate features from datasets that might not be possible to do with classical computers [9]. This makes QML a promising tool for drug discovery, optimization, and other areas of science and engineering [2].

Despite its potential, QML is still in its infancy, and many challenges must be overcome to develop practical QML algorithms and hardware. One of the most significant challenges is developing a fault-tolerant quantum computer. The current hardware of quantum computers falls under the category of noisy intermediate-scale quantum (NISQ), where we have a few hundred noisy qubits. As a result, running complex quantum algorithms on NISQ devices can be challenging [20]. Another challenge is developing efficient algorithms that can take advantage of the limited qubit resources of current quantum computers. Hence, researchers are trying to figure out algorithms suitable for NISQ devices [3].

In this paper, we implement a hybrid ML model that combines the classical ML model and variational quantum circuit (VQC). VQCs are parameterized quantum circuits that can learn parameters based on optimization methods and are helpful for many applications, including ML [16]. We applied this model to the MNIST dataset, which consists of numerical digits from 0 to 9. The model consists of a classical convolutional neural network (CNN) model combined with a VQC in the final layer that will be used for the prediction. The parameters of the quantum circuit are trained in a similar way to the weights in a classical model. This approach enabled us to effectively train the model as well as classify images with high accuracy. Furthermore, this hybrid model requires only a few qubits and shallow circuit depths, making it suitable for NISQ hardware. This hybrid model can serve as a basis for future QML applications and research.

2 Preliminary

2.1 Quantum Computing

In traditional classical computing, information is processed and stored as bits, which are either 0 or 1. These bits are used to represent information in the form of digital signals that are processed by classical computers. However, in quantum computing, the basic unit of information is the qubit (quantum bit), which is a two-state quantum-mechanical system that can exist in a superposition of both 0 and 1 states simultaneously [17]. A qubit can be represented by a vector in two-dimensional complex Hilbert space $\mathbf{C}^2$, and its superposition form can be written as

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

where $|0\rangle = [1 \quad 0]^T$ and $|1\rangle = [0 \quad 1]^T$ which represents the orthonormal basis for $\mathbf{C}^2$ known as the computational basis and $\alpha, \beta \in \mathbf{C}$ are the probability amplitudes obeying $|\alpha|^2 + |\beta|^2 = 1$.

A quantum gate is a fundamental operation that can be performed on a quantum system, such as a qubit, in quantum computing. These gates are denoted by a unitary operator $\mathbf{U}$ in the Hilbert space. They serve as the quantum equivalent of classical logic gates in traditional computing and enable manipulation of the qubit's state. The Pauli gates (X, Y, and Z) and the Hadamard gate are examples of single qubit quantum gates. A Hadamard gate transforms the qubit to a superposition state that is equally likely to be either 0 or 1. The Hadamard operation can be represented with matrix notation as

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2}$$

Another type of single qubit quantum gates are the quantum rotation gates, which rotates the qubit state around the X, Y, and Z axes. These gates are typically represented by unitary matrices and are defined by a rotation angle. There are three common type of rotation gates namely $\mathbf{R_x}(\theta)$, $\mathbf{R_y}(\theta)$, and $\mathbf{R_z}(\theta)$ that are given in computational basis by

$$\mathbf{R_x}(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\iota\sin\frac{\theta}{2} \\ -\iota\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \mathbf{R_y}(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, \mathbf{R_z}(\theta) = \begin{bmatrix} e^{-\iota\frac{\theta}{2}} & 0 \\ 0 & e^{\iota\frac{\theta}{2}} \end{bmatrix}. \tag{3}$$

To achieve a comprehensive set of gates for quantum computation, it is essential to include two-qubit gates. The Controlled NOT (CNOT) gate is one of the most commonly used two-qubit gates, which flips the state of the target qubit if the control qubit is in the state $|1\rangle$ and leaves it unchanged if the control qubit is in the state $|0\rangle$. The CNOT gate is represented in the computational basis as follows

$$\mathbf{C_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

Once the information in the qubits has been processed by quantum gates, the result of the computation must be obtained through quantum measurement. Quantum measurement is achieved by projecting the quantum state onto the basis of the Hilbert space. In the case of a single qubit using the computational basis, the measurement operators $\mathbf{M}_0 = |0\rangle\langle 0|$ and $\mathbf{M}_1 = |1\rangle\langle 1|$ correspond to measurement outcomes 0 and 1 respectively.

2.2 Variational Quantum Circuits (VQCs)

There are many quantum circuits that can implement various quantum algorithms. One class is the VQCs that are used for optimizing the parameters of a quantum algorithm to obtain a specific output [4]. In a VQC, the quantum gates used in the circuit are parameterized, meaning that the values of the gate parameters can be adjusted to obtain a desired output. The circuit is initialized in a specific quantum state, and the parameters of the circuit are iteratively updated using a classical optimization algorithm until the desired output is obtained. An example of VQC is shown in Fig. 1. VQCs are particularly useful in situations where the exact solution to a problem is difficult to obtain using classical methods. By optimizing the parameters of the quantum circuit, the output of the circuit can be used as an approximation to the exact solution. VQCs have been demonstrated to be capable of solving problems in quantum chemistry, optimization, and ML with a level of accuracy that surpasses classical methods in certain cases [5,16,26].

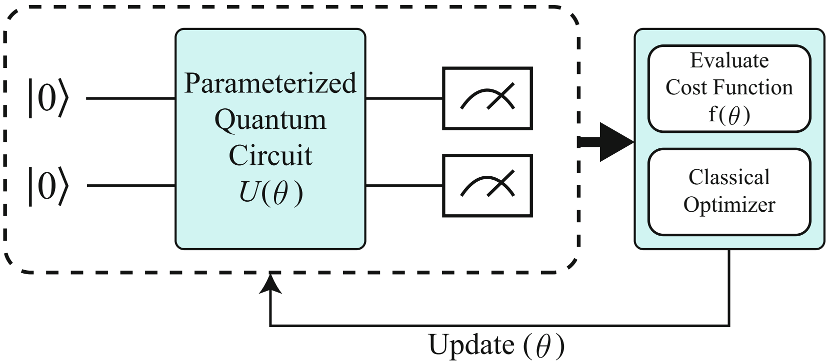

Fig. 1. A variational quantum circuit.

2.3 Neural Networks

A neural network is a type of ML technique that employs layers of interconnected nodes or neurons to process data [29]. Each neuron has an associated weight and threshold and is connected to other neurons. When the output of any neuron exceeds the specified threshold, the neuron is activated by an activation function, and the data is passed to the next layer. Otherwise, no data is transmitted to the next layer of the network. Through a kind of machine perception, the neural network can label or cluster raw input data. These networks learn similarly to the human brain by analyzing labeled or unlabeled training examples [1].

During the training process of a neural network, the weights are initialized randomly. Following this, forward propagation takes place where the input data is fed into the neural network, and the activation of neurons in each layer is computed using the current weights and biases. The neural network's loss is then determined by measuring the error between the predicted and actual labels. To update the weights and biases in the neural network, we employ backward propagation to compute the gradient of the loss function. The gradient is then used to adjust the weights and biases in the opposite direction, aiming to minimize the loss function. Finally, the weights and biases in the neural network are updated using an optimization algorithm such as Adam or gradient descent in the parameter update rule. This process is repeated until the neural network has converged and the loss function has reached its minimum. Once the neural network is trained, its performance on unseen data is evaluated through a test set. With the advent of deep learning, neural networks have emerged as a powerful technology in recent years, finding numerous applications across different domains. For instance, they are used in image and speech recognition, as well as natural language processing (NLP) [18,22,27].

In image processing, CNNs are very popular [14]. They use specialized layers such as convolutional, pooling, activation, batch normalization, dropout, and fully connected layers. The convolutional layer performs convolution operations on the input image to extract features. The pooling layer reduces the spatial dimensions of the input by downsampling, typically using max or average pooling. The activation layer applies a non-linear activation function such as ReLU to introduce non-linearity into the network. The batch normalization layer normalizes the input data to speed up training and improve model stability. The dropout layer randomly drops out units from the network during training to reduce overfitting. The fully connected layer performs classification or regression on the output of the preceding layers. By stacking these layers together, CNNs can learn hierarchical representations of the input data and make predictions based on them.

3 Methods

We used a hybrid quantum-classical model which enables the use of quantum computers along with classical computers. This approach allows us to use a small number of qubits and circuit depth by outsourcing some computation to classical computers. On the classical side, we used CNN to process the input image and provide meaningful features [12,13]. On the quantum side, we used VQC to process the extracted features and classify the given image into the correct label [15]. We used Hadamard, parameterized rotation, and CNOT gates for our VQC. After the VQC computation, we measured the quantum state and used the measurement outcomes to predict the input images. From the loss function, we used the backward propagation algorithm to update the parameters in the VQC as well as the weights in CNN. To process classical data with a quantum circuit, we need to first encode the data from a classical one into a quantum domain. This encoding enables the mapping of the classical data into high-dimensional Hilbert space with a nonlinear mapping. In this high-dimensional Hilbert space, the quantum operation is performed to move the data position, enabling easier classification compared to lower dimensional space. There are several ways to realize this encoding process, such as basis encoding, Hamiltonian encoding, and angle encoding [24]. We used angle encoding as the classical data can be easily used as angles in rotation gates.

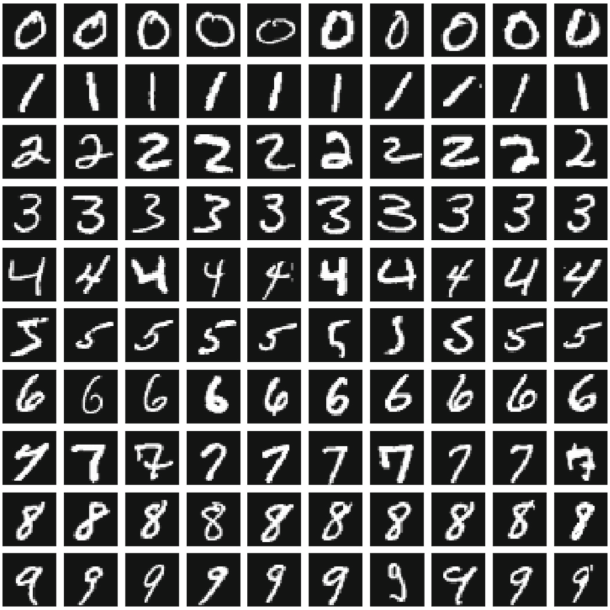

Fig. 2. Images from MNIST dataset.

We used MNIST handwritten digits dataset to train and test our hybrid model shown in Fig. 2. The dataset has images of digits from 0–9, resulting in a 10-classes classification problem. We used 5000 images from MNIST dataset where the images are divided with a ratio of 8:2 into training and testing datasets. The training dataset is divided again with the same ratio into training and validation datasets. The size of the input image is 28×28 pixels with a single channel. These input images are fed into the CNN that act as a feature extractor which is followed by the VQC as the predictor. Block diagram of the hybrid model is shown in Fig. 3.

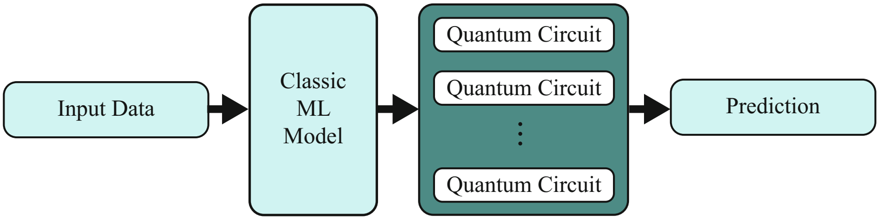

Fig. 3. Hybrid quantum-classical ML model.

For the hybrid model, we first used a convolutional layer with 5×5 kernel size and 16 filters followed by a max-pooling layer with strides of 2 on its output.

Then, we used a convolutional layer with the same kernel size and 32 filters. The next convolutional layer used 3×3 kernel size with 64 filters and followed by a max-pooling layer with strides of 2. Then, a two-dimensional dropout layer is applied. The output is flattened and fed to the fully connected layer having 128 neurons. The convolutional and fully connected layers used the ReLU activation function. The CNN model architecture is summarized in Fig. 4. Then, we again used a fully connected layer with $10 \times n_p$ where n_p is the number of parameters used in the quantum circuit. This output of the fully connected layers is fed to the quantum circuit.

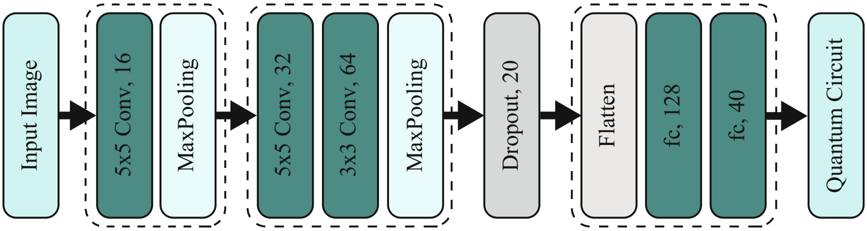

Fig. 4. CNN model architecture.

For our VQC, we used a 3 qubits quantum circuit and 4 parameters. We initialized the circuit with an equal superposition state by means of Hadamard gates. This Hadamard operation can be represented with matrix notation as,

$$(\mathbf{H}|0\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle, \tag{5}$$

where n is the number of qubits. Then, a parameterized rotation around Z axis is applied to every qubit which can be represented as,

$$\bigotimes_{i=1}^{n} \mathbf{R_z}(\phi_i), \tag{6}$$

where n is the number of qubits. We applied an entangling gate, CNOT gate, between qubit 3 and 2. The circuit is followed by a parameterized rotation around the Z axis on qubit 2 and a CNOT gate between qubit 2 and 1. We then applied Hadamard gate to qubit 1. Our final unitary gate can be represented as

$$\mathbf{U}(\phi) = (\mathbf{H} \otimes \mathbf{I} \otimes \mathbf{I})(\mathbf{C_x}^{(2)} \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{R_z}(\phi_4) \otimes \mathbf{I})(\mathbf{I} \otimes \mathbf{C_x}^{(3)}) \bigotimes_{i=1}^{3} \mathbf{R_z}(\phi_i)\mathbf{H}, \tag{7}$$

where we used $\mathbf{C_x}^{(k)}$ to denote the CNOT gate with qubit k as the control qubit and $\mathbf{I}$ denotes the identity matrix. Finally, we measured the first qubit using the computational basis, which can result in outcomes 0 and 1 with probability,

$$p(i) = \langle \psi(\phi)|\mathbf{M}_i \otimes \mathbf{I} \otimes \mathbf{I}|\psi(\phi)\rangle, \tag{8}$$

where $|\psi(\phi)\rangle = \mathbf{U}(\phi)|0\rangle^{\otimes 3}$ and $i = \{0, 1\}$. We then classified the input image based on the measurement outcome. Our VQC is shown in Fig. 5.

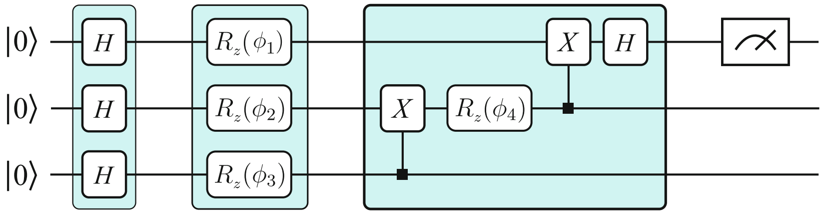

Fig. 5. VQC of our hybrid quantum-classical neural network.

We used 10 such quantum circuits where each quantum circuit corresponds to each class. In general, quantum circuits have the ability to learn complex features which might not be possible classically [9]. We took the probability of getting outcome 0 for each quantum circuit and predicted that the input image is in class p if the p-th quantum circuit has the maximum probability among all the circuits. The quantum circuit is trained using gradient that is computed using the parameter-shift rule [28]. For each parameter in the quantum circuit, we can calculate the gradient as,

$$\partial_{\phi_i} \mathbf{U}(\phi) = \mathbf{U}(\phi + \epsilon \cdot \mathbf{e_i}) - \mathbf{U}(\phi - \epsilon \cdot \mathbf{e_i}), \tag{9}$$

where ϵ is the shift of the parameter and $\mathbf{e_i}$ is the i-th column of $n_p \times n_p$ identity matrix.

We implemented the quantum circuit using Qiskit 0.38.0 with 'qasm simulator'. Qiskit is an open-source software development kit for quantum computers [21]. We used PyTorch 1.12.1 to apply the CNN model. We trained the model for 20 epochs and using cross-entropy as the loss function. The optimizer was set as Adam with a learning rate of 0.001. We used 1000 shots for the quantum measurement and set $\epsilon = \pi/2$ for the parameter-shift rule.

4 Results

To evaluate the performance of the hybrid model, we calculated the precision, recall, F1-score, and accuracy of the model, which are the commonly used performance metrics [19]. These performance metrics are defined as

$$\text{precision} = \frac{t_p}{t_p + f_p}, \tag{10}$$

$$\text{recall} = \frac{t_p}{t_p + f_n}, \tag{11}$$

$$\text{F1} - \text{score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{12}$$

$$\text{accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}, \tag{13}$$

where t_p, t_n, f_p, and f_n are the number of true positive, true negative, false positive, and false negative cases, respectively. The results are shown in Table 1. We can see that the model performed well in all the categories with an average accuracy of 97%. The validation loss and accuracy plot can be seen in Fig. 6, where the model converges as the number of epochs increase.

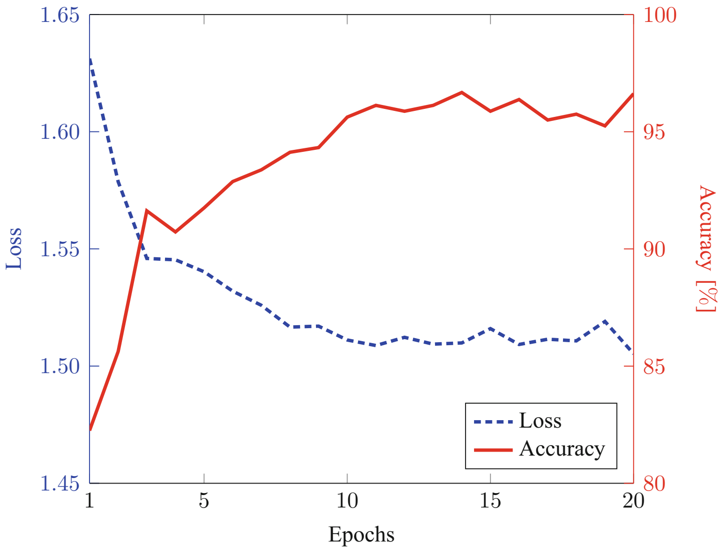

Fig. 6. The validation loss and accuracy of the model are plotted as a function of epochs.

We also provided the confusion matrix on the test dataset in Fig. 7 where we can see that the hybrid model can predict the actual digits of the test dataset images with at least 95% correct prediction. These results showed that quantum and classical computers could be trained together to perform machine learning tasks. Specifically, the VQC can play a role as a predictor in image classification task with good performance with the help of CNN as the feature extractor.

Table 1. Classification report of our hybrid model on MNIST dataset.

Digit	Precision	Recall	F1-score
0	0.99	0.98	0.98
1	0.99	0.98	0.99
2	0.97	0.96	0.96
3	0.98	0.95	0.97
4	0.96	0.96	0.96
5	0.99	0.99	0.99
6	0.99	1.00	0.99
7	0.98	0.98	0.98
8	0.98	0.98	0.98
9	0.91	0.95	0.93
Accuracy	N/A	N/A	0.97
Macro Avg	0.97	0.97	0.97

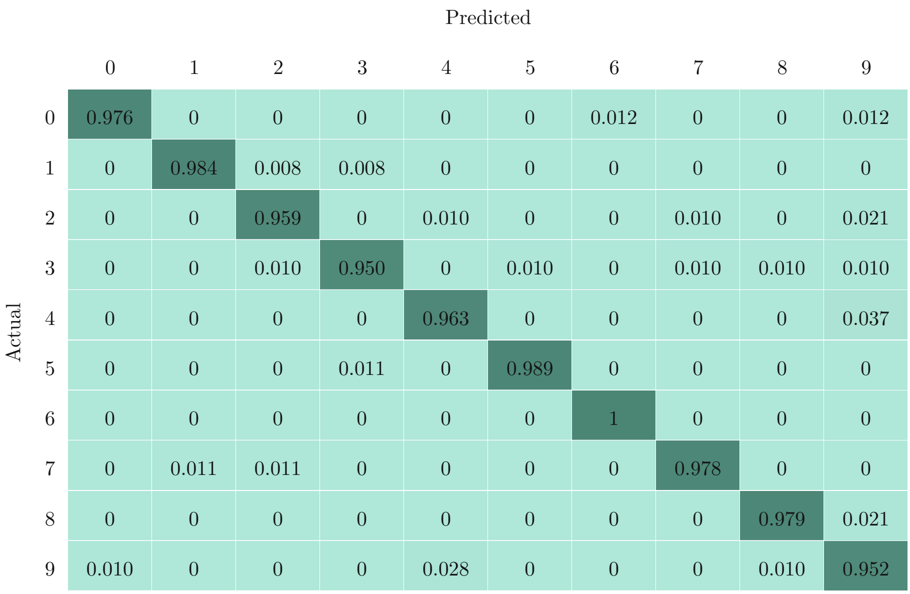

Fig. 7. Confusion matrix of hybrid model on test dataset.

5 Conclusion

QML is an interdisciplinary field that combines quantum physics and machine learning algorithms to solve complex problems more efficiently. In this study, we demonstrated a hybrid approach that utilizes a conventional CNN model with a VQC for image classification with good accuracy. This approach paves the way for quantum-enhanced machine learning, which aims to leverage quantum mechanical effects to enhance machine learning performance. Additionally, this hybrid VQC and QML algorithm can be adapted to the limitations of NISQ devices by adjusting circuit parameters, such as depth and number of qubits. This flexibility allows researchers to explore the potential of hybrid algorithms, even in the face of hardware limitations.

References

1. Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A., Arshad, H.: State-of-the-art in artificial neural network applications: a survey. Heliyon **4**(11), e00938 (2018)
2. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
3. Boixo, S., et al.: Characterizing quantum supremacy in near-term devices. Nat. Phys. **14**(6), 595–600 (2018)
4. Cerezo, M., et al.: Variational quantum algorithms. Nat. Rev. Phys. **3**(9), 625–644 (2021)
5. Chen, S.Y.C., Yang, C.H.H., Qi, J., Chen, P.Y., Ma, X., Goan, H.S.: Variational quantum circuits for deep reinforcement learning. IEEE Access **8**, 141007–141024 (2020)
6. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep. Prog. Phys. **81**(7), 074001 (2018)
7. Duong, T.Q., Ansere, J.A., Narottama, B., Sharma, V., Dobre, O.A., Shin, H.: Quantum-inspired machine learning for 6G: fundamentals, security, resource allocations, challenges, and future research directions. IEEE Open J. Veh. Technol. **3**, 375–387 (2022)
8. Duong, T.Q., Nguyen, L.D., Narottama, B., Ansere, J.A., Huynh, D.V., Shin, H.: Quantum-inspired real-time optimization for 6g networks: opportunities, challenges, and the road ahead. IEEE Open J. Commun. Soc. **3**, 1347–1359 (2022)
9. Havlíček, V., et al.: Supervised learning with quantum-enhanced feature spaces. Nature **567**(7747), 209–212 (2019)
10. Huang, H.Y., et al.: Power of data in quantum machine learning. Nat. Commun. **12**(1), 2631 (2021)
11. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS 2012, vol. 1, pp. 1097–1105. Curran Associates Inc., Red Hook (2012)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

14. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J.: A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans. Neural Netw. Learn. Syst. **33**, 6999–7019 (2021)
15. McClean, J.R., Romero, J., Babbush, R., Aspuru-Guzik, A.: The theory of variational hybrid quantum-classical algorithms. New J. Phys. **18**(2), 023023 (2016)
16. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**(3), 032309 (2018)
17. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information. Am. J. Phys. **70**(5), 558–559 (2002)
18. Palaz, D., Magimai-Doss, M., Collobert, R.: End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. Speech Commun. **108**, 15–32 (2019)
19. Powers, D.M.W.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv arXiv:2010.16061 (2018)
20. Preskill, J.: Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
21. Qiskit contributors: Qiskit: an open-source framework for quantum computing (2023)
22. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: a survey. Sci. Chin. Technol. Sci. **63**(10), 1872–1897 (2020)
23. Rizvi, S.M.A., Asif, N., Ulum, M.S., Duong, T.Q., Shin, H.: Multiclass classification of metrologically resourceful tripartite quantum states with deep neural networks. Sensors **22**(18), 6767 (2022)
24. Schuld, M., Killoran, N.: Quantum machine learning in feature Hilbert spaces. Phys. Rev. Lett. **122**(6), 040504 (2019)
25. Schuld, M., Sinayskiy, I., Petruccione, F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172–185 (2014)
26. Tilly, J., et al.: The variational quantum eigensolver: a review of methods and best practices. Phys. Rep. **986**, 1–128 (2022)
27. Traore, B.B., Kamsu-Foguem, B., Tangara, F.: Deep convolution neural network for image recognition. Ecol. Inform. **48**, 257–268 (2018)
28. Wierichs, D., Izaac, J., Wang, C., Lin, C.Y.Y.: General parameter-shift rules for quantum gradients. Quantum **6**, 677 (2022)
29. Zou, J., Han, Y., So, S.S.: Overview of artificial neural networks. Artif. Neural Netw. Meth. Appl. **458**, 14–22 (2009)