Challenges in Hybrid Quantum-Classical Convolutional Neural Networks: Evaluating Qubit Impact on Binary Image Classification

Van-Nui Nguyen[1*], *Vu-Hai* Nguyen[2**], *Quang-Quy* Tran[1***], *Duc-Binh* Nguyen[1****], and *Tran-Chung* Dao[1†]

[1]Thai Nguyen University of Information and Communication Technology
[2]Department of Computer Science and Information Engineering, National Chung Cheng University

Abstract. Machine learning (ML) has proven its efficacy in acquiring information from previous data and leveraging that understanding to handle practical problems. The recent development of quantum computing, based on the principles of quantum physics, has made the integration of quantum computing and machine learning a prominent area of research. Nonetheless, the practical utilization of quantum computing in machine learning continues to present considerable obstacles, especially regarding implementation and the management of real-world tasks. This research presents a hybrid quantum-classical convolutional neural network model (HQ-CNN) and evaluates its performance on binary image classification tasks using subsets of the MNIST and EMNIST datasets. The study also presents an in-depth discussion of the challenges that arise from integrating quantum computing into machine learning, emphasizing the training procedure, the results obtained post-training, and the number of qubits used in the research. The experimental findings indicate that increasing the number of qubits generally enhances image classification efficacy. However, an increased number of qubits does not inherently ensure enhanced overall model performance.

1 Introduction

Artificial Intelligence (AI) covers a wide range of methodologies for constructing systems capable of imitating human intelligence, including cognition, behavior, and perception in computers. Although the term AI was first introduced in the 1950s, its applications have proliferated in recent decades, particularly within the domain of machine learning (ML), where computers demonstrate the capacity to proactively learn and enhance their performance without specific programming [1]. ML has been applied to many domains and address real-world problems in the forth industrial revolution such as manufacturing industries [2] and different fields.

With the introduction of Quantum Computing (QC) in 1982 [3], it leverages the features of quantum mechanics, including superposition and entanglement, to represent and handle data. Both concepts allow quantum computers to resolve specific complicated problems considerably more rapidly than traditional computers nowadays [4]. QC benefits enterprises by providing faster solutions to complex combination problems, facilitating advancements in cryptography, chemistry, finance, and manufacturing, even prior to the availability of fully mature quantum machines [5]. This inspires an investigation into the integration of quantum computing and machine learning to utilize the advantages of both domains—namely, enhanced computational capacity, accelerated processing speed, and enhanced efficacy in addressing the growing scale and complexity of problems prevalent in the current data-driven world. However, Noisy intermediate-scale quantum (NISQ) is a term by John Preskill in 2018 [6], refers to the current quantum computers that are defined by significant error rates and limited in scale by the number of logical or physical qubits present in the system. Xiaozhen et al. proposed landscape-based methodologies [7] to enhance hybrid quantum-classical algorithms on NISQ's devices, addressing the fundamental limitations of the NISQ era. Their work mainly focuses on optimization in noisy quantum contexts, while also establishing a foundation for broader progress in hybrid quantum-classical machine learning, especially in a context where quantum hardware and algorithms encounter considerable obstacles.

Recently, the hybrid quantum-classical machine learning methodology has drawn an increased research and application projects producing encouraging outcomes. Namely, Ovalle-Magallanes and other authors [8] proposed a Hybrid Classical-Quantum Convolutional Neural Network (HQ-CNN) for the detection of X-ray coronary angiography and achieved accuracy, recall and F1-score at 91.8033%, 94.9153%, and 91.8033%, respectively. In binary image classification, Deepak et al. [9] selected two labels, 0 and 1, from the MNIST dataset and introduced a Hybrid Quantum-Classical Neural Network (H-QNN) model, their proposed model achieved an accuracy of 99.7%, which is significantly higher than that of other classical models. In the context of Earth observation data, Fan and co-authors [10] conducted a benchmark

*e-mail: nvnui@ictu.edu.vn
**e-mail: g13410151@ccu.edu.tw
***Corresponding author. e-mail: tqquy@ictu.edu.vn
****e-mail: ndbinh@ictu.edu.vn
†e-mail: dtchung@ictu.edu.vn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

study using datasets such as Overhead-MNIST, So2Sat LCZ42, PatternNet, RSI-CB256, and NaSC-TG2 through the TensorFlow Quantum platform. Their findings indicated that the hybrid quantum–classical convolutional neural network (HQ-CNN) surpassed classical models in performance. Despite the positive outcomes reported, previous research investigations have not thoroughly examined the effects of adjusting qubit numbers in HQ-CNN models, nor have they conducted comprehensive performance analyses under diverse quantum configurations. Furthermore, the findings typically fail to evaluate the performance of HQ-CNNs in comparison to classical CNNs utilizing almost identical model designs. We hypothesis the performance of the HQ-CNN model if it were designed to closely resemble the conventional CNN architecture. We investigate the influence of qubit quantity on the performance of HQ-CNN regarding assessment metrics and training duration. To maintain objectivity in assessing the impact of different qubit counts, the training is performed on both the MNIST and EMNIST dataset — an aspect that distinguishes our work from prior studies, which typically analyze performance on a single dataset only. In context of the current progress and growing impact of research on hybrid quantum–classical convolutional neural networks (HQ-CNNs), we present the following major contributions in this paper:

1. A Hybrid Quantum Convolutional Neural Network (HQ-CNN) model is proposed as a benchmark for the MNIST and EMNIST datasets, which demonstrates how the number of qubits affects both the predictive performance and computational efficiency of the HQ-CNN. We investigate the outcomes and compare them to those from a classical CNN model.

2. We analyze the challenges highlighted by the experimental findings, especially in training HQ-CNN models for practical applications. Additionally, we also provide a source code[1] for further implementation and research.

2 Methodology

2.1 Data Preparation

In this work, we utilize the MNIST [11] and EMNIST [12] datasets to benchmark the HQ-CNN model. Considering that this is a binary classification problem, we chose suitable class pairs from each dataset. We selected the digits 0 and 1 from the MNIST dataset, which has handwritten digits ranging from 0 to 9. For the EMNIST dataset, an expanded version of MNIST containing both letters and digits, we chose the letters A and B from the *Letter* split, which consists of handwritten English alphabets mapped to 26 uppercase classes (A-Z) for the classification task. Both MNIST and EMNIST were derived from the NIST Special Database 19 [13].

[1] https://github.com/tnvuhai/Challenges-in-Hybrid-Quantum-Classical-Convolution-Neural-Network-Evaluating-Qubit-Impact-on-Image-.git

Table 1. Comparison of original and binary subsets of MNIST and EMNIST datasets

Dataset	Training		Testing		Ratio of Binary
	Original	Binary	Original	Binary	(Training:Testing)
MNIST	50,000	12,665	10,000	2,115	≈ 6 : 1
EMNIST (letters)	88,800	9,600	14,800	1,600	≈ 6 : 1

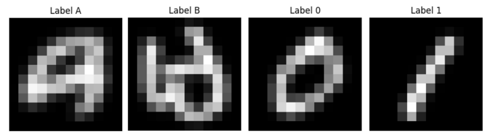

Figure 1. EMNIST dataset (Label A&B) and MNIST dataset (Label 0&1)

In Table 1, Original indicates the entire dataset containing all labels as specified in the original source. For the binary classification task, only the previously mentioned labels were selected. The quantity of samples associated with these designated labels is documented under Binary. The table also presents the estimated ratio of training and testing data used for the binary task following splitting from the original dataset.

Prior to model training, all input images underwent preprocessing using a three-step pipeline. Initially, each image was transformed to grayscale to maintain uniformity in the channel count. The images were then scaled to a uniform resolution of 12×12 pixels to standardize the input dimensions for the model, as can be seen in Figure 1 .The photos were eventually transformed into tensor format and normalized to the $[0, 1]$ range, which additionally restructured the data into a channel-first format ($C \times H \times W$) as necessitated by PyTorch-based models later on.

2.2 Fundamental Architecture

The architecture of hybrid quantum-classical model consists of two components: the classical layer, the quantum layer. They collaborate to tackle two-class classification issues utilizing both classical and quantum computing methodologies. This type of architecture has been utilized in relevant studies [8–10]. The simplified structure is depicted in Figure 2:

As can be seen in the Figure 2, h_n is the value of parameters in neural network in each hidden layer, and $R(h_n)$ is rotation gates that rotates by h_n. The final prediction value from the HQ-CNN is presented by y, a sigmoid function ranging from $(0, 1)$:

Classical Layer: The classical layer refers to the traditional components of convolutional neural networks (CNNs) [14] and performs two main functions within the hybrid design. Initially, in the input-to-hidden phase, the classical layer consists of convolutional and fully connected layers, which enable feature extraction and allow suitable application of backpropagation for updating model parameters. Subsequently, the obtained features pass through to the quantum layer, where quantum computation is executed utilizing a Variational Quantum Circuit (VQC). After the quantum computation concludes,

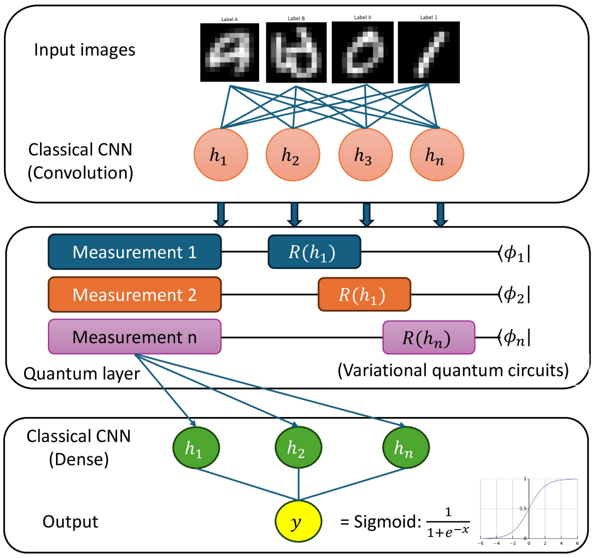

Figure 2. Architecture of Hybrid Quantum-Classical Neural network model

the primary function of the quantum circuit is to collapse quantum states into classical information through measurement, therefore allowing the classical layer to proceed to the following stage of computation., which then applies a dense layer to produce the final output.

Quantum Layer: The quantum layer serves as a connector between classical and quantum processing by encoding classical data into quantum states to facilitate quantum computation. Upon completion of the computation, the quantum states are collapsed through measurement, changing them to classical data for the following processing and final output. The quantum layer has two primary components: the Variational Quantum Circuit (VQC) and the measurement procedure. The subsequent components will be further discussed in Section 3.

3 HQ-CNN architecture and baselines

3.1 Experimental setup and hyperparameter configuration

We present a pipeline for evaluating the HQ-CNN model and compare it with a classical CNN, in accordance with the aims of this paper. The following is a detailed explanation of the main components of the pipeline:

1. **Data Preparation**: This phase is elaborated in Section 2.1, which outlines the preparation of the MNIST and EMNIST datasets for training objectives.

2. **Model Training and Evaluation**: The training data is utilized to train the models employing the hyperparameters specified in this section. Upon completion of training, the testing data is instantly input into the model to calculate assessment metrics such as accuracy, precision, F1-score, binary cross-entropy loss, and training time.

3. **Repeated Training and Evalution**: The training procedure in step 2 is reiterated five times. The final results can be obtained as the mean and standard deviation of the evaluation metrics throughout these iterations, offering a comprehensive evaluation of model efficacy.

The study employed two models for benchmarking: proposed HQ-CNN and classical CNN, both trained under identical hardware configurations, support libraries, and model parameter settings. This research was performed utilizing a conventional physical computing system with quantum simulation. The complete configuration is outlined below.

- **CPU**: Intel(R) Xeon(R) Silver 4210R, featuring 20 cores and 40 threads.
- **GPU**: NVIDIA RTX A6000 with 60GB of VRAM.
- **RAM**: 252 GB

We employed the `PennyLane` library (version 0.38.0) to integrate quantum computing with machine learning for the implementation of both the HQ-CNN and standard CNN models. We utilized `PyTorch` [15] version 2.6.0+cu124 for the classical CNN model. PennyLane shows significant compatibility with PyTorch, hence simplifying the implementation and evaluation of both models within a unified framework.

The training procedure for both models utilized the following hyperparameters: a batch size of 16, 5 training epochs, and a consistent learning rate of 0.01. For the HQ-CNN model, we employed four distinct qubit configurations — [2, 4, 6, 8] — in the quantum layer for evaluation.

3.2 Quantum Layer

We proposed a custom Quantum Layer ultilizing Variational Quantum circuits (VQC), which use a classical optimizer to train a parameterized quantum circuits (PQCs) to integrate quantum computing into our hybrid model, this architecture conforms to the fundamental principles of quantum computing. Quantum layer operates deploying the Pennylane library with a PyTorch-compatible QNode interface, facilitating seamless incorporation into traditional deep learning frameworks. The Quantum Layer takes classical inputs $x_i \in \mathbb{R}^n$ and encodes them into quantum circuits using **rotation gates**. Each input element is employed to rotate a singular qubit via both the **RX** and **RZ** gates, as mentioned in equation 1 and 2:

$$RX(\theta) = \exp\left(-i\frac{\theta}{2}X\right) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \quad (1)$$

$$RZ(\phi) = \exp\left(-i\frac{\phi}{2}Z\right) = \begin{bmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{i\phi/2} \end{bmatrix} \quad (2)$$

These operations execute rotations of the qubit state around the X and Z axes of the Bloch sphere, therefore encoding the classical input x_i into a quantum state.

The circuit then implements a BasicEntanglerLayers template from Pennylane, a commonly utilized variational ansatz for facilitating entanglement between qubits, as it can be defined in quation 3, while RY and CNOT gates are defined in [**?**]:

$$U_{\text{entangle}}(\boldsymbol{\theta}) = \prod_{l=1}^{L}\left(\bigotimes_{i=1}^{n} RY(\theta_{l,i}) \cdot \text{CNOT}_{i,i+1}\right) \quad (3)$$

The entanglement structure is important for capturing correlations among variables and modeling complex functions. We defines $\theta = \pi x_i$ and the utilization of πx_i instead of simply x_i is a prevalent strategy to enhance the diversity of quantum states from the input, hence creating more different variations among various input data.

The final phase of the circuit involves measurements. Measurement is the technique of obtaining classical information from a quantum state. A solitary qubit is represented by a vector in Hilbert space, it can be defined by a waveform ψ in equation 4:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where } |\alpha|^2 + |\beta|^2 = 1 \quad (4)$$

In this context, α and β indicate complex probability amplitudes, with their squared magnitudes $|\alpha|^2$ and $|\beta|^2$ representing the probabilities of measuring the qubit in the states $|0\rangle$ and $|1\rangle$, respectively. Each qubit is measured in the Z-basis, yielding the expected value of the Pauli-Z operator in equation 5:

$$\langle Z_i \rangle = \langle \psi | Z_i | \psi \rangle = |\alpha|^2 - |\beta|^2 \quad (5)$$

The expected value lies in the interval [-1, 1], yielding a real-valued classical quantity applicable in further computations.

3.3 HQ-CNN and CNN models

We propose the HQ-CNN and classical CNN models in accordance with the principles defined in [14]. Both models are executed using PyTorch, with the HQ-CNN further utilizing the Pennylane library to incorporate quantum layers. Table 2 indicates the complete structures of the two models. Both models begin with a convolutional and pooling layer for spatial feature extraction, succeeded by flattening and fully connected layers. The HQ-CNN model integrates a quantum processing component (QuantumLayer) between the feature extraction and classification phases to augment the representational capacity using a variational quantum circuit.

The primary distinction, undoubtedly, lays in the existence of the quantum layer. Before accessing the quantum layer in the HQ-CNN model, a Tanh activation function is utilized instead of the ReLU function employed in the conventional CNN. This selection is driven by the characteristic of the Tanh function, which produces outputs inside the interval $[-1, 1]$, a crucial aspect for quantum gates like RX and RY that utilize rotation angles measured in radians. Utilizing ReLU prior to the quantum layer in HQ-CNN would induce a periodicity effect due to unbounded outputs, potentially preventing the model's convergence.

Table 2. Architectural Comparison between Classical CNN and HQ-CNN

Layer Type	**CNN Model**	**HQ-CNN Model**
Input Layer	$1 \times 12 \times 12$ grayscale image	$1 \times 12 \times 12$ grayscale image
Convolutional Layer	Conv2D($1 \to 12$ filters, kernel size 3×3) + ReLU	Conv2D($1 \to 12$ filters, kernel size 3×3) + ReLU
Pooling Layer	MaxPool2D (kernel size 2×2)	MaxPool2D (kernel size 2×2)
Flatten Layer	Flatten()	Flatten()
Fully Connected Layer	Linear($300 \to 52$) + ReLU	Linear($300 \to n\_qubits$) + Tanh
Quantum Processing	None	QuantumLayer(n_qubits)
Output Layer	Linear($52 \to 1$) + Sigmoid	Linear($n\_qubits \to 1$) + Sigmoid

4 Results

4.1 Evaluation metrics

The performance of the proposed HQ-CNN model and CNN baseline is evaluated using established evaluation metrics, as outlined in this section. These metrics are computed based on the following terms:

- **True Positive (TP):** the number of accurately predicted positive instances.
- **True Negative (TN):** the number of accurately predicted negative instances.
- **False Positive (FP):** the number of negative instances incorrectly classified as positive.
- **False Negative (FN):** the number of positive instances incorrectly classified as negative.

Using these definitions, the evaluation metrics are computed as follows:

Accuracy: evaluates the overall accuracy of the model across every class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Precision: Indicates the proportion of correctly predicted true positive samples among all predicted positive samples.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

Recall: measures the ratio of accurately predicted positive cases to the total number of actual positive cases.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

F1-score: the harmonic mean of precision and recall presents a balanced statistic, particularly advantageous for imbalanced datasets:

$$F1 - Score = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

Binary Cross Entropy Loss (BCELoss): Utilized as the loss function for binary classification tasks. It measures the differences between the predicted probability and the actual binary labels:

$$BCELoss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \quad (10)$$

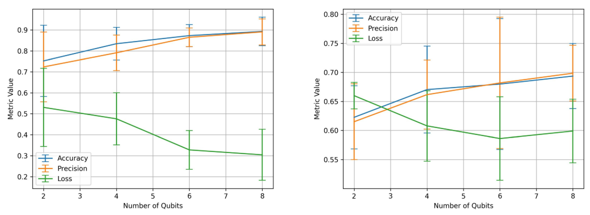

Figure 3. HQ-CNN performance vs Number of Qubits in MNIST (left) and EMNIST (right)

where $y_i \in \{0, 1\}$ is the true label, and $\hat{y}_i \in (0, 1)$ is the predicted probability.

4.2 Model Performance and Discussion

This section provides a comparison of the proposed Hybrid Quantum Convolutional Neural Network (HQ-CNN) and the baseline Classical CNN in the task of image classification, which are tested on MNIST and EMNIST. The models were trained and evaluated under uniform conditions to ensure an unbiased comparison, with performance evaluated using standard metrics: accuracy, precision, F1-score, and loss.

Table 3 presents a comparative examination of HQ-CNN models utilizing varying quantities of qubits, in conjunction with the traditional CNN baseline. The results indicate an undeniable trend: Traditional CNN consistently surpasses all HQ-CNN versions in the handwritten character classification job across both the MNIST and EMNIST datasets, according to our proposed design.

This observation emphasizes that even a relatively simple CNN can attain robust performance, whereas hybrid quantum-classical models may underperform due to existing technology constraints. Specifically, during the Noisy Intermediate-Scale Quantum (NISQ) era, inadequately designed quantum components like HQ-CNN may not only show poor performance but also not take full advantage on the prospective benefits of quantum computing due to their inability to manage quantum noise during training.

Present implementations continue to depend significantly on current simulators, which are physically slower and computationally costly. The practical implementation of HQ-CNN is limited by the constraints of current hardware infrastructure, primarily reliant on classical computing devices.

Figure 3 shows that increasing the number of qubits significantly enhances the performance of the HQ-CNN model in all evaluation metrics. This enhancement is due to the increased expressive capability of the quantum circuit: when additional qubits are incorporated, the system may represent information in a higher-dimensional Hilbert space, thus capturing more complex data characteristics and correlations. More specific, although the total number of parameters in the HQ-CNN model increases from 2 to 8 qubits, it remains significantly lower than that of the classical CNN, highlighting the potential of quantum computing (QC) for future development. Specifically, within the Quantum Layer, 2–8 qubits correspond to 2–8 parameters in the quantumLayer(n_qubits) configuration shown in Table 2. This demonstrates that even a minimal change in the number of parameters can significantly impact the results—an effect not observed in traditional models.

This advantage, however, is accompanied by significant constraints. Increasing the quantity of qubits substantially extends training duration, particularly in classical simulation contexts, owing to the exponential escalation in quantum state complexity. Furthermore, in the existing NISQ (Noisy Intermediate-Scale Quantum) hardware environment, the incorporation of additional qubits frequently results in heightened vulnerability to quantum noise, rendering training less robust and more prone to errors. As observed in the results, the standard deviation across all outputs is notably large and fluctuating, executing these models on NISQ devices causes instability, contrary to earlier successful HQ-CNN studies. Frequent challenges involve overfitting, vanishing gradients, and hard-to-optimize variational quantum circuits with multiple parameters. These issues highlight the necessity for improved quantum circuit design, noise-resistant training methodologies, and, eventually, superior quantum hardware to fully leverage the advantages of hybrid quantum-classical models.

5 Conclusion

This study explores how varying the number of qubits affects the performance of a Hybrid Quantum-Classical Convolutional Neural Network (HQ-CNN) on classification tasks. Results show that performance generally improves with more qubits, up to a point, beyond which gains diminish due to increased circuit complexity and gradient instability. Configurations with 4–6 qubits offer the best trade-off between accuracy, efficiency, and training stability in NISQ-era simulations. While classical CNNs still outperform HQ-CNNs in tasks like digit classification, HQ-CNNs exhibit potential advantages in quantum feature representation under certain conditions. These findings highlight the importance of balancing circuit depth and qubit count for practical hybrid model deployment.

Furthermore, the analysis indicates that excessive qubit expansion may lead to barren plateau phenomena, thereby hindering effective gradient-based optimization during training. Future research should investigate adaptive circuit design strategies and noise-aware optimization techniques to enhance scalability and robustness of HQ-CNN architectures on near-term quantum hardware. In addition, a comprehensive evaluation across diverse datasets and real quantum processors is necessary to rigorously assess generalization capability, hardware-induced noise sensitivity, and the practical feasibility of scaling HQ-CNN models beyond simulated NISQ environments.

6 Acknowledgements

This work was funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant No. 102.05-2023.49.

Table 3. Comparison of CNN and HQ-CNN on MNIST and EMNIST datasets. Metrics reported as mean ± std.

Dataset	Model	Qubits	Accuracy	Precision	F1-score	Loss	Total parameters
MNIST	HQ-CNN	2	0.7522 ± 0.1698	0.7237 ± 0.1662	0.8245 ± 0.1099	0.5305 ± 0.1863	727
		4	0.8341 ± 0.0780	0.7908 ± 0.0846	0.8647 ± 0.0535	0.4760 ± 0.1243	1333
		6	0.8772 ± 0.0527	0.8648 ± 0.0445	0.8838 ± 0.0481	0.3279 ± 0.0922	1939
		8	0.8927 ± 0.0680	0.8905 ± 0.0621	0.9007 ± 0.0644	0.3044 ± 0.1216	2545
	CNN	–	0.9987 ± 0.0003	0.9978 ± 0.0000	0.9988 ± 0.0003	0.0033 ± 0.0028	15825
EMNIST	HQ-CNN	2	0.6226 ± 0.0543	0.6150 ± 0.0651	0.6520 ± 0.0313	0.6599 ± 0.0228	727
		4	0.6703 ± 0.0746	0.6616 ± 0.0594	0.6625 ± 0.1166	0.6078 ± 0.0605	1333
		6	0.6798 ± 0.1126	0.6819 ± 0.1130	0.6838 ± 0.1048	0.5861 ± 0.0719	1939
		8	0.6936 ± 0.0560	0.6985 ± 0.0478	0.6850 ± 0.0748	0.5990 ± 0.0547	2545
	CNN	–	0.9862 ± 0.0038	0.9841 ± 0.0091	0.9863 ± 0.0037	0.0451 ± 0.0095	15825

References

[1] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen, Y. Mei, Y. Sun, B. Tran, B. Xue, M. Zhang, A survey on evolutionary machine learning, Journal of the Royal Society of New Zealand **49**, 205 (2019). 10.1080/03036758.2019.1609052

[2] P. Larrañaga, D. Atienza, J. Diaz-Rozo, A. Ogbechie, C. Puerto-Santana, C. Bielza, Industrial applications of machine learning (CRC Press, 2018)

[3] R.P. Feynman, in *Feynman and computation* (cRc Press, 2018), pp. 133–153

[4] R. Rietsche, C. Dremel, S. Bosch, L. Steinacker, M. Meckel, J.M. Leimeister, Quantum computing, Electronic Markets **32**, 2525 (2022).

[5] F. Bova, A. Goldfarb, R.G. Melko, Commercial applications of quantum computing, EPJ quantum technology **8**, 2 (2021).

[6] J. Preskill, Quantum computing in the nisq era and beyond, Quantum **2**, 79 (2018). 10.22331/q-2018-08-06-79

[7] X. Ge, R.B. Wu, H. Rabitz, The optimization landscape of hybrid quantum–classical algorithms: From quantum control to nisq applications, Annual Reviews in Control **54**, 314 (2022). https://doi.org/10.1016/j.arcontrol.2022.06.001

[8] E. Ovalle-Magallanes, J.G. Avina-Cervantes, I. Cruz-Aceves, J. Ruiz-Pinales, Hybrid classical–quantum convolutional neural network for stenosis detection in x-ray coronary angiography, Expert Systems with Applications **189**, 116112 (2022). https://doi.org/10.1016/j.eswa.2021.116112

[9] D. Ranga, S. Prajapat, Z. Akhtar, P. Kumar, A.V. Vasilakos, Hybrid quantum–classical neural networks for efficient mnist binary image classification, Mathematics **12** (2024). 10.3390/math12233684

[10] F. Fan, Y. Shi, T. Guggemos, X.X. Zhu, Hybrid quantum-classical convolutional neural network model for image classification, IEEE Transactions on Neural Networks and Learning Systems **35**, 18145 (2024). 10.1109/TNNLS.2023.3312170

[11] Y. LeCun, C. Cortes, C. Burges, Mnist handwritten digit database, ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist **2** (2010).

[12] G. Cohen, S. Afshar, J. Tapson, A.V. Schaik, Emnist: Extending mnist to handwritten letters, 2017 International Joint Conference on Neural Networks (IJCNN) (2017). 10.1109/ijcnn.2017.7966217

[13] P.A. Flanagan, NIST handprinted forms and characters - NIST special database 19 (2016)

[14] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: Analysis, applications, and prospects, IEEE Transactions on Neural Networks and Learning Systems **33**, 6999 (2022). 10.1109/TNNLS.2021.3084827

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., Pytorch: An imperative style, high-performance deep learning library (2019), `1912.01703`, `https://arxiv.org/abs/1912.01703`