Classical-Quantum CNN Hybrid for Image Classification

Muhamad Akrom[1], Wahyu Aji Eko Prabowo[2]
[1,2]Computer Science Faculty, Universitas Dian Nuswantoro, Semarang, Indonesia
E-mail: [1]m.akrom@dsn.dinus.ac.id, [2]prabowo@dsn.dinus.ac.id

Abstract

This study explores Quantum Convolutional Neural Network (QCNN) starting from foundational quantum operations, such as the Rx gate for encoding MNIST image data into quantum states. We implemented quantum convolutional and pooling layers using one_unitary and two_unitary circuits, enabling effective feature extraction and dimensionality reduction while preserving critical information. Expressibility analysis revealed varying capabilities across different one_unitary circuits, with Rx, Ry, and Rz combinations demonstrating promising results akin to Haar random states. The proposed QCNN model exhibited robust performance metrics (accuracy: 95.98%, precision: 94.44%, recall: 96.59%, F1-score: 0.9551, AUC: 0.9604) in classification tasks, supported by efficient convergence during optimization. Future directions include expanding QCNN applications to handle more complex datasets and optimizing architectures to enhance quantum machine learning capabilities, particularly in image processing. This study underscores the potential of QCNNs in advancing quantum computing applications in neural network architectures.

Keywords: MNIST, classification, CNN, expressibility

1. INTRODUCTION

Quantum computing takes advantage of the non-classical characteristics of quantum states in a wide range of applications; it has grown in popularity [1], [2], [3], [4], [5], [6]. While certain activities have no classical equivalents, others have considerable quantum benefits [7], [8], [9]. Entanglement and superposition are two fundamental ideas that have fundamentally changed how information is represented in quantum computing. Unlike traditional computers, quantum computers use qubits rather than bits. Due to the intrinsic complexity of qubits, the basic units of information, quantum computers can do tasks beyond the capabilities of regular systems. Due to their entanglement and superposition, qubits may perform computational tasks faster than traditional computers, a significant departure from the field [10], [11], [12], [13].

Thanks to superposition, qubits can concurrently exist in states that represent 0 and 1. This property allows quantum computers to explore several possibilities at once and carry out intricate computations [14], [15], [16], [17], [18]. A qubit's quantum state is demonstrated by the equation $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, emphasizing its capacity to exist in several states concurrently. This sets quantum computing apart from classical paradigms and further demonstrates the superior processing power of quantum computers. In a two-dimensional complex Hilbert space, the quantum state of a qubit is represented by the basis vectors $|0\rangle$ and $|1\rangle$. The superposition of the qubit in the states $|0\rangle$ and $|1\rangle$ is determined by the coefficients α and β. A crucial concept that describes the interdependence between the states of several qubits is entanglement, which enables quantum computers to perform exceptionally well while processing large datasets and solving certain computational problems [19], [20], [21].

Regression and data classification are two areas where machine learning (ML) works well [22], [23], [24], [25]. Nevertheless, it is difficult to apply quantum concepts to machine learning. A developing topic that combines quantum computing and machine learning is called quantum circuit learning (QCL) or quantum machine learning (QML) [26], [27]. With different quantum gate operations, QML algorithms may be constructed as quantum circuits [28], [29]. With major benefits over conventional approaches, particularly for near-term quantum devices, QML has the potential to propel quantum computing in the noisy intermediate-scale quantum (NISQ) future [29], [30]. Quantum convolutional neural network (QCNN) is an integral part of the QML paradigm that utilizes the principles of quantum mechanics to improve processing and pattern recognition capabilities [31], [32].

Classical CNN primarily employs convolution layers to extract features from images [33], [34]. However, in QCNN, these convolution layers are instantiated through quantum circuits. Convolution is achieved by applying quantum gates to the image representation. The image representation can be encoded as a quantum state vector, wherein each element corresponds to an image pixel. Within the framework of QCNN, superposition enables concurrent processing of information across multiple pixel states, while entanglement facilitates the association of pertinent image features. Moreover, QCNN can harness additional quantum principles, such as amplitude probability modeling, to ascertain optimal weights during the learning phase [35], [36], [37]. The advent of QCNN heralds a significant breakthrough in machine learning, as it integrates quantum concepts into a well-established image classification architecture. Despite being in its nascent stages of exploration, QCNN holds immense promise in enhancing the efficacy and efficiency of classification systems, particularly in managing intricate image datasets such as MNIST.

In this paper, we introduce the concept of an adaptive Quantum Convolutional Neural Network (QCNN) architecture tailored for the classification of the MNIST dataset. A distinguishing feature of our research lies in utilizing quantum circuits as the foundational framework for our architecture, which seamlessly integrates into the Quantum Machine Learning (QML) paradigm. We explore how fundamental quantum principles, including superposition, entanglement, and amplitude probability modeling, can be effectively harnessed to enhance the task of image classification. Through comprehensive experimentation and analysis, we assess the efficacy and practical viability of the adaptive QCNN architecture in effectively handling the intricacies of the MNIST dataset. Consequently, this paper endeavors to furnish novel insights into the advancement of pioneering and efficient classification algorithms within the realm of QML.

2. METHOD

In this work, we implemented the Qiskit module on the IBM Quantum platform to build a Python programming language for quantum operations to present experiments conducted on the MNIST dataset for binary classification of digits 0 and 1.

1.1. Dataset and pre-processing

The MNIST (Modified National Institute of Standards and Technology) dataset is a comprehensive database of handwritten digits widely utilized for training various image processing systems. The MNIST dataset ensures completeness by providing 70,000 images of handwritten digits, ranging from 0 to 9 (Figure 1a). Among these, 60,000 images constitute the training set, while 10,000 images form the testing set. These images are presented in grayscale with a standard resolution of 28×28 pixels. The image pixel values range from 0, representing black, to 255, representing white. To enhance computational efficiency, the images in the dataset were normalized to pixel values between 0 and 1. Subsequently, the data was filtered to retain only images of the digits 0 and 1 for binary classification (Figure 1b). This resulted in a reduced subset comprising 8,000 training images and 2,000 testing images [38], [39], [40]. The images were further flattened into one-dimensional arrays (equivalent to 768 pixels) to facilitate efficient processing by the model. Principal Component Analysis (PCA) is applied to reduce the dimensionality of data in optimizing processing by QCNN. In quantum computing, each qubit can be seen as a representation of the data dimension. Thus, reducing the dimensionality from 768 pixels to 4 pixels significantly reduces the number of qubits needed, making quantum processing more feasible and efficient. These preprocessing steps ensure the efficient execution of code while preserving data integrity, with this focused subset designed to accelerate experimental workflows [41], [42], [43].

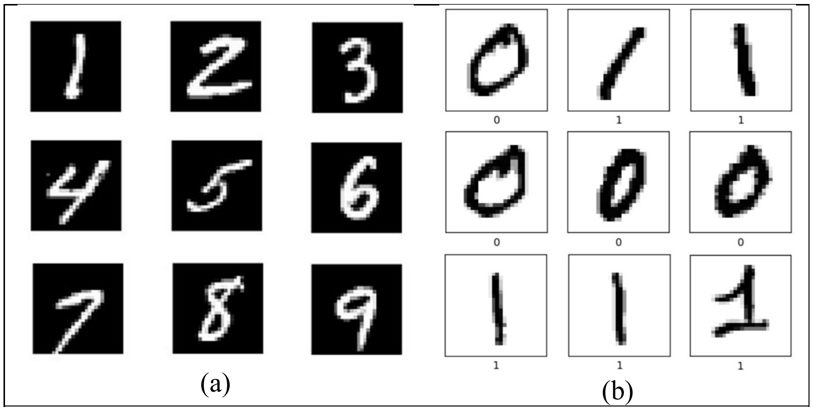
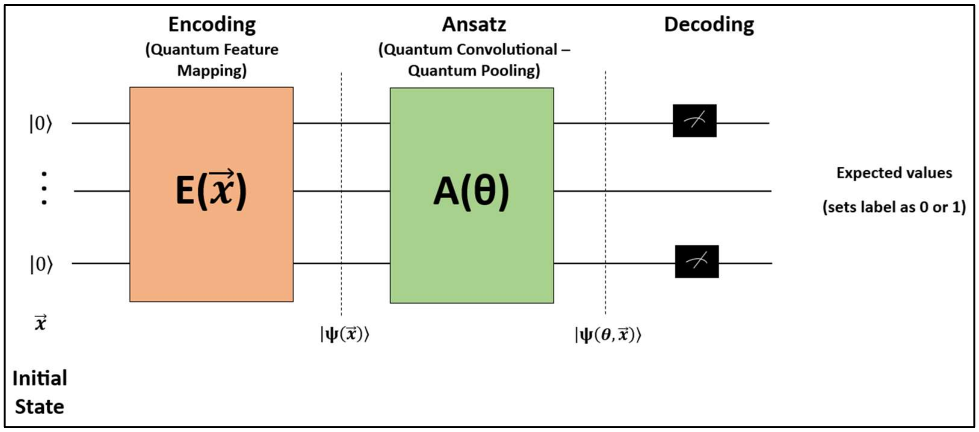


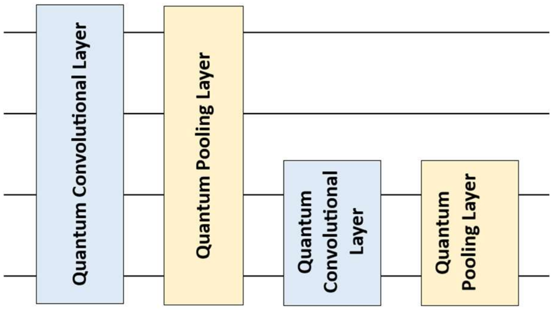
Figure 1. Examples of dataset before and after pre-processing

1.2. Adaptive QCNN Modeling

Integrating quantum-based programming into adaptive QCNN requires recalibrating to make it executable on a quantum computer. The proposed work aims to investigate the potential of QCNN for classification on the MNIST dataset. The suggested QCNN's design is seen in Figure 1.



(a)



(b)

Figure 2. The architecture of the proposed (a) adaptive QCNN and (b) quantum convolutional-pooling layers.

From Figure 2a, the proposed QCNN architecture comprises three main layers: encoding, ansatz, and decoding. Each layer plays a crucial role in quantum processing. This architecture aims to leverage the principles of convolutional neural networks in the quantum domain to provide classification tasks. The encoding layer is responsible for transforming classical data into quantum states. This process is known as quantum feature mapping. Classical data is encoded into the rotation angles of quantum gates applied to qubits. The encoded quantum state is the input for the ansatz layers of the QCNN. The ansatz layer (Figure 2b) consists of quantum convolutional layers and quantum pooling layers designed to extract and summarize features from the quantum data. Quantum convolutional layers apply parameterized quantum gates to subsets of qubits, similar to how convolutional filters operate on local regions of classical data. The goal is to extract local features by applying a series of unitary transformations. Quantum pooling layers reduce the number of qubits, analogous to pooling layers in classical CNNs that reduce the spatial dimensions of data. By stacking multiple convolutional and pooling layers, the QCNN can progressively extract higher-level features while reducing the dimensionality of the quantum state. The decoding layer transforms the processed quantum state into a form that can be interpreted classically. This typically involves measuring the qubits and applying classical post-processing to the measurement outcomes.

Assigning qubits to each feature in the data is the initial stage in the QCNN technique. After that, each feature is mapped, often known as encoding. In this fundamental stage, a quantum state is constructed in the quantum Hilbert space using quantum feature mapping [44], [45]. This method entangles and superpositions qubits in a variety of states. By applying it to the ground state $|0\rangle$ for each qubit, the quantum feature mapping, $\mathrm{E}(\vec{x})$, as given in equation 1, encodes the classical input vector $\vec{x}$ into the quantum state vector $|\psi(\vec{x})\rangle$ for each observation. All the classical properties are essentially related to a single qubit.

$$\mathrm{E}(\vec{x})|0\rangle = |\psi(\vec{x})\rangle \tag{1}$$

The QCNN performs classification by assessing the degree of similarity between data vectors in the quantum feature space using an ansatz denoted by $\mathrm{A}(\theta)$ in equation 2, following the initial encoding of classical properties into quantum states [46], [47].

$$\mathrm{A}(\theta)|\psi(\vec{x})\rangle = |\psi(\theta, \vec{x})\rangle \tag{2}$$

After the classifier has been trained, binary value encodings convert quantum state data into conventional values of 0 and 1 [48], [49]. Additionally, parameterized (θ) are altered to lower the loss function.

1.3. Model Assessment

ROC curves, recall, F1-score, accuracy, and precision are crucial when evaluating classifier models. Accuracy is defined as the ratio of precise forecasts to overall predictions. While accuracy calculates the proportion of true positive forecasts among all positive forecasts, recall assesses the percentage of true positive predictions among all real positives. The F1-score balances these two metrics, the harmonic mean of recall and accuracy. ROC curves plot the true positive rate against the false positive rate, showing performance across thresholds [50], [51].

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \tag{3}$$

$$Precision = \frac{TP}{(TP+FP)} \times 100 \tag{4}$$

$$Recall\ (Sensitivity) = \frac{TP}{(TP+FN)} \times 100 \tag{5}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{(Precision + Rec\quad)} \tag{6}$$

Here, TP stands for true positives, TN for true negatives, FP for false positives, and FN for false negatives. These metrics provide vital information about how effectively the model can reliably classify examples. They do this by extensively evaluating the model's functionality and anticipated accuracy inside a classification framework.

1.4. Parameter Optimization

QCNN adaptively develops encoding and ansatz parameters to maximize data processing using unique circuit architectures. This method improves data categorization and strengthens the model's learning and classification skills by assessing the similarity between data pairs in the quantum feature space. Gradient optimization uses methods like ADAM, COBYLA, and AQGD to modify the model parameters and minimize the loss function. Quantum models improve their performance in classification problems by iterative computing and updating the gradient, which results in more informative encodings and ideal ansatz structures.

3. RESULTS and DISCUSSION

In the initial phase of our investigation, we employed simple quantum operations for encoding and constructing the quantum convolutional and pooling layers. Based on Figure 3, we utilized the rotation gate Rx for the encoding layer. The choice of this gate for encoding was driven by its simplicity and effectiveness in transforming classical data into a quantum state. By applying the Rx gate to each qubit, we encoded the normalized pixel values of the MNIST images into rotation angles, thereby embedding the classical data into the quantum domain. The quantum convolutional and quantum pooling layers were constructed using a combination of one_unitary and two_unitary circuits. Figure X depicts the one_unitary circuit consisting of a sequence of single-qubit rotations, Rx, Ry, and Rz gates. Combining these gates, the one_unitary circuit allows for flexible and comprehensive manipulation of single-qubit states. The two_unitary circuit builds on the one_unitary circuit by incorporating controlled-NOT (CX) and controlled-Z gates. This enhancement enables entanglement between qubits, which is crucial for capturing correlations within the data. The quantum convolutional layers applied these circuits to subsets of qubits, effectively extracting local features from the quantum state. Integrating one_unitary and two_unitary circuits provided a balance between single-qubit operations and entanglement, enhancing the convolutional layers' expressive power. The quantum pooling layer was designed to reduce the dimensionality of the quantum state while retaining essential information. Measurements followed this reduction to determine the significant qubits, ensuring the retention of critical data features.

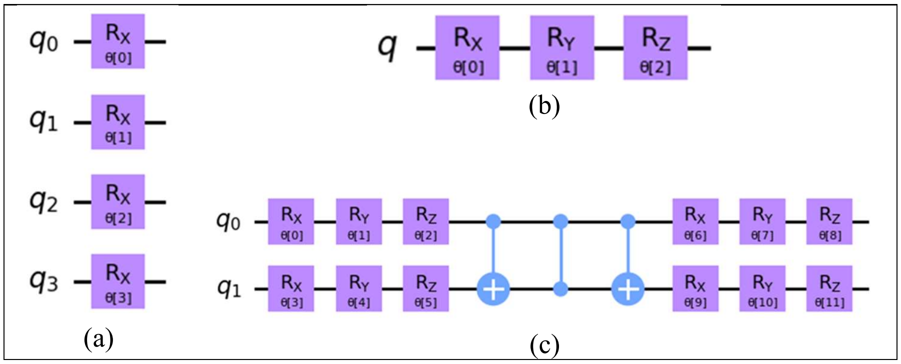


Figure 3. Simple circuit example of (a) encoding, (b) one_unitary, and (c) two_unitary

The expressibility value quantifies how effectively the circuit can generate diverse quantum states across its parameter space. To evaluate the expressibility of quantum circuits, the Haar measure must define a uniform distribution across unitary operators. A maximally expressible Haar random state evenly samples the Hilbert space. The efficiency of quantum circuits in navigating the quantum state space is measured by comparing them to the Haar measure. Therefore, we tested the combination of gate sequences that make up the one_unitary circuit, considering that this circuit is the main basis for forming the entire QCNN circuit. Table 1 shows that (a) Rz, Ry, and Rz combination gates are the most powerful; they produce a wide state distribution similar to Haar random states, which makes them the best choice for variational quantum algorithms. The expressibility values across different one_unitary circuits (a) to (f) vary significantly, indicating differences in their ability to explore quantum state space. Higher expressibility values suggest that the circuit can access a broader range of quantum states, which can be advantageous in capturing complex features in quantum data. In the context of QCNNs, choosing one_unitary circuit directly influences the network's capability to extract and process features from quantum data. Circuits with lower expressibility values (such as (a) Rx, Ry, Rz with 0.09228) may offer enhanced representational power, potentially improving classification accuracy and robustness. Figure 4 shows that (a) Rx, Ry, and Rz combinations are perfect for one_unitary circuits because they offer a uniform state distribution on the Bloch sphere and best resemble the Haar measure, comparing diverse combinations.

Table 1. Expressibilty value of one_unitary circuits

	One_unitary circuit	Expressibility
(a)	Rx, Ry, Rz	0.09228
(b)	Rx, Rz, Ry	0.11508
(c)	Ry, Rx, Rz	0.09238
(d)	Ry, Rz, Rx	0.12227
(e)	Rz, Rx, Ry	0.18761
(f)	Rz, Ry, Rx	0.18918

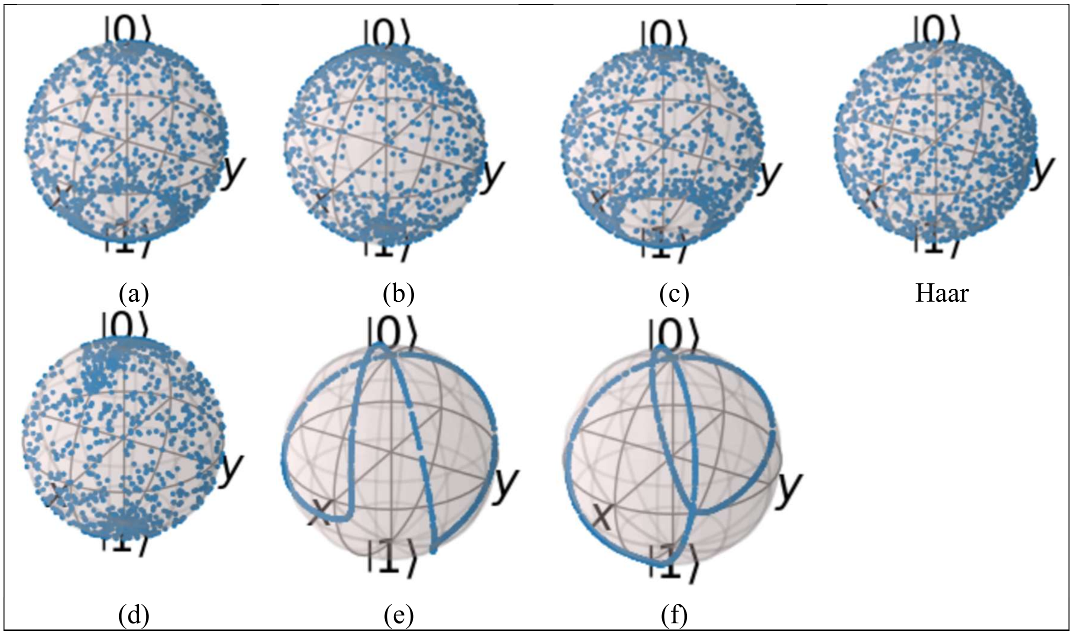


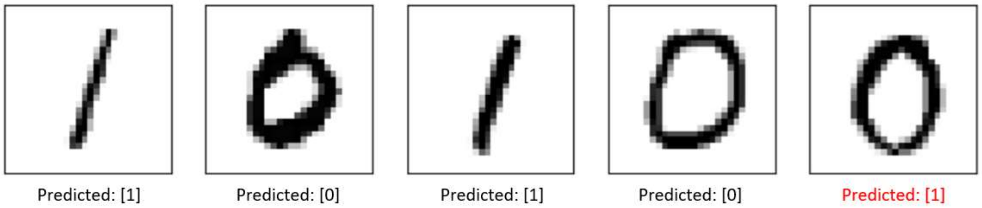
Figure 4. Bloch's expressibility representation based on Haar measurement for variational circuits

The performance metrics of the QCNN model, as summarized in Table 2, exhibit strong capabilities in classification tasks. The model achieves an accuracy of 95.98%, indicating its proficiency in correctly predicting both positive and negative classes. Precision, measuring the proportion of correctly predicted positive instances among all predicted positives, is 94.44%, underscoring the model's precision in positive predictions. Moreover, the recall metric, reflecting the proportion of actual positives correctly identified by the model, demonstrates a high value of 96.59%, highlighting the model's ability to capture true positive instances effectively. The F1-score, which harmonizes precision and recall into a single metric, is calculated at 0.9551, suggesting a balanced performance in achieving precision and recall objectives. Furthermore, the Area Under the Curve (AUC) score of 0.9604 indicates the excellent discriminatory ability of the model in distinguishing between positive and negative classes based on its predicted probabilities. The QCNN model exhibits robust performance across various evaluation metrics, affirming its suitability for classification tasks. These metrics underscore the model's reliability in practical applications in this work.

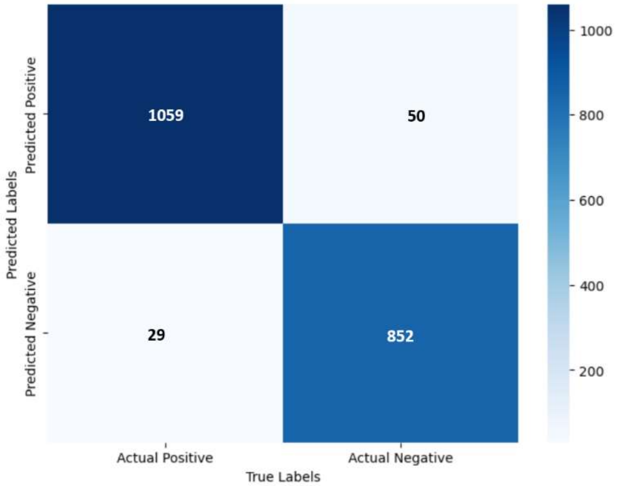
Table 2. QCNN model performances

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC**
QCNN	0.9598	0.9444	0.9659	0.9551	0.9604

Figure 5a is an example of the predicted results given the actual values. The confusion matrix (Figure 5b) summarizes these results, showing the number of TP, FP, TN, and FN. The confusion matrix is crucial for understanding a classification model's performance. It quantifies prediction outcomes relative to class labels, detailing model accuracy and errors. In the matrix, rows represent true classes, and columns represent predicted instances. Correctly classified instances are off-diagonal, where predictions match actual outcomes, while misclassifications appear on the diagonal. Metrics like TP, TN, FP, and FN are delineated. For example, a matrix shows 1060 TP, 50 FP, 30 FN, and 850 TN. Perfect classification is indicated by only diagonal elements, suggesting accurate performance across classes. This tool not only clarifies alignment between predictions and outcomes but also identifies areas for model improvement, enhancing overall effectiveness.



(a)



(b)

Figure 5. (a) Examples of prediction values and (b) confusion matrix of QCNN

The convergence analysis depicted in Figure 6 illustrates the onset of the convergence phase of the model around iteration 150, providing insights into the QCNN's optimization process and its efficacy in minimizing the objective function (or cost function). Initially, the QCNN undergoes an exploratory phase exploring diverse parameter configurations and settings. The objective function values exhibit notable fluctuations during this stage as the model adjusts to varying inputs and weight distributions. As the iterations progress into the later phases, there is a discernible trend toward stabilization and reduction in the objective function values. Importantly, the absence of prolonged plateau periods suggests a continuous, smooth decline in the objective function values, indicative of consistent progress and steady convergence. This observation underscores the QCNN's ability to systematically refine its parameters and optimize them to achieve lower objective function values, enhancing its performance in quantum machine learning applications.

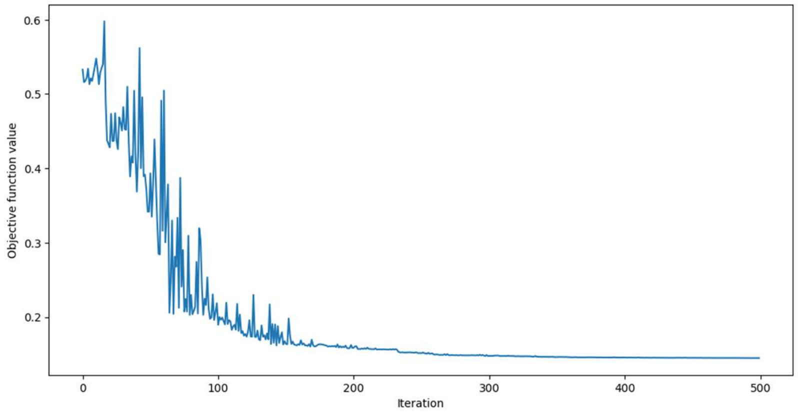


Figure 6. A plot of the objective function value (cost function) of QCNN

The internal quantum circuit architecture will determine how well the learning model performs in performance evaluation. Although the simulations in the experimental findings are microscopic, we would like to use the QCNN model on more complex data for future work. More comprehensive QCNN techniques and implementations. Furthermore, by encoding more information into a single qubit using the QCNN model, much more effective learning can be achieved in image processing. In the future, we will use various simulation techniques to examine the QCNN model, which is more effective and has superior learning performance.

4. CONCLUSION

Our study began with simple quantum operations like the Rx gate for encoding MNIST image data into quantum states. We constructed quantum convolutional and pooling layers using one_unitary and two_unitary circuits, enabling effective feature extraction and dimensionality reduction while preserving critical information. Analysis of expressibility across different one_unitary circuits highlighted variations in their ability to explore quantum state spaces, with Rx, Ry, and Rz combinations showing promising results akin to Haar random states. The proposed QCNN model demonstrated strong performance metrics (accuracy: 95.98%, precision: 94.44%, recall: 96.59%, F1-score: 0.9551, AUC: 0.9604) in classification tasks, supported by effective convergence during optimization. Looking forward, our focus remains on expanding QCNN applications to complex datasets and refining architectures for enhanced quantum machine learning performance, particularly in image processing.

REFERENCES

[1] Biamonte, J., Wittek, P., Pancotti, N. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017). https://doi.org/10.1038/nature23474

[2] Palittapongarnpim, P ., Wittek, P . & Sanders, B. C. Controlling adaptive quantum phase estimation with scalable reinforcement learning. In *Proc. 24th Eur. Symp. Artificial Neural Networks (ESANN-16) on Computational Intelligence and Machine Learning* 327–332 (2016)

[3] Scherer, A. et al. Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target. *Quantum Inform. Process.* **16**, 60 (2017)

[4] E. Grant *et al.*, "Hierarchical quantum classifiers," *npj Quantum Inf*, vol. 4, no. 1, Dec. 2018, doi: 10.1038/s41534-018-0116-9.

[5] Wiebe, N., Kapoor, A. & Svore, K. M. Quantum perceptron models. *Adv. Neural Inform. Process. Syst.* **29**, 3999–4007 (2016)

[6] Wiebe, N. & Granade, C. Can small quantum systems learn? Preprint at https://arxiv.org/abs/1512.03145 (2015)

[7] Low, G. H., Yoder, T. J. & Chuang, I. L. Quantum inference on Bayesian networks. *Phys. Rev. A* **89**, 062315 (2014)

[8] Rønnow, T. F. et al. Defining and detecting quantum speedup. *Science* **345**, 420–424 (2014)

[9] Schuld, M., Fingerhuth, M. & Petruccione, F. Quantum machine learning with small-scale devices: implementing a distance-based classifier with a quantum interference circuit. Preprint at https://arxiv.org/abs/1703.10793 (2017)

[10] Wittek, P. & Gogolin, C. Quantum enhanced inference in Markov logic networks. *Sci. Rep.* **7**, 45672 (2017)

[11] Denchev, V. S., Ding, N., Matsushima, S., Vishwanathan, S. V. N. & Neven, H. Totally corrective boosting with cardinality penalization. Preprint at https://arxiv.org/abs/1504.01446 (2015)

[12] O'Gorman, B. A. et al. Bayesian network structure learning using quantum annealing. *EPJ Spec. Top.* **224**, 163–188 (2015)

[13] Denchev, V. S ., Ding, N ., Vishwanathan, S . & Neven, H. Robust classification with adiabatic quantum optimization. In *Proc. 29th Int. Conf. on Machine Learning (ICML-2012)* (2012)

[14] Tezak, N. & Mabuchi, H. A coherent perceptron for all-optical learning. *EPJ Quant. Technol.* **2**, 10 (2015)

[15] Cai, X.-D. et al. Entanglement-based machine learning on a quantum computer. *Phys. Rev. Lett.* **114**, 110504 (2015)

[16] Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017)

[17] Lovett, N. B., Crosnier, C., Perarnau-Llobet, M. & Sanders, B. C. Differential evolution for many-particle adaptive quantum metrology. *Phys. Rev. Lett.* **110**, 220501 (2013)

[18] Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **109**, 050505 (2012)

[19] August, M. & Ni, X. Using recurrent neural networks to optimize dynamical decoupling for quantum memory. Preprint at https://arxiv.org/abs/1604.00279 (2016)

[20] Banchi, L., Pancotti, N. & Bose, S. Quantum gate learning in qubit networks: Toffoli gate without time-dependent control. *npj Quant. Inf.* **2**, 16019 (2016)

[21] Las Heras, U., Alvarez-Rodriguez, U., Solano, E. & Sanz, M. Genetic algorithms for digital quantum simulations. *Phys. Rev. Lett.* **116**, 230504 (2016).

[22] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[23] Zahedinejad, E., Ghosh, J. & Sanders, B. C. High-fidelity single-shot Toffoli gate via quantum control. *Phys. Rev. Lett.* **114**, 200502 (2015)

[24] Dolde, F. et al. High-fidelity spin entanglement using optimal control. *Nat. Commun.* **5**, 3371 (2014)

[25] Wiebe, N., Granade, C. & Cory, D. G. Quantum bootstrapping via compressed quantum Hamiltonian learning. *New J. Phys.* **17**, 022005 (2015)

[26] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys Rev Lett*, vol. 113, no. 3, Sep. 2014, doi: 10.1103/PhysRevLett.113.130503.

[27] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp Phys*, vol. 56, no. 2, pp. 172–185, Apr. 2015, doi: 10.1080/00107514.2014.964942.

[28] Granade, C. E., Ferrie, C., Wiebe, N. & Cory, D. G. Robust online Hamiltonian learning. *New J. Phys.* **14**, 103013 (2012)

[29] Ventura, D. & Martinez, T. Quantum associative memory. *Inf. Sci.* **124**, 273–296 (2000)

[30] H. Y. Huang *et al.*, "Power of data in quantum machine learning," *Nat Commun*, vol. 12, no. 1, Dec. 2021, doi: 10.1038/s41467-021-22539-9.

[31] Kieferova, M. & Wiebe, N. Tomography and generative data modeling via quantum Boltzmann training. Preprint at https://arxiv.org/abs/1612.05204 (2016)

[32] Chowdhury, A. N. & Somma, R. D. Quantum algorithms for Gibbs sampling and hitting-time estimation. *Quant. Inf. Comput.* **17**, 41–64 (2017)

[33] Temme, K., Osborne, T. J., Vollbrecht, K. G., Poulin, D. & Verstraete, F. Quantum metropolis sampling. *Nature* **471**, 87–90 (2011)

[34] Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning. *Phys. Rev. A* **94**, 022308 (2016).

[35] Denil, M . & De Freitas, N. Toward the implementation of a quantum RBM. In *Neural Information Processing Systems (NIPS) Conf. on Deep Learning and Unsupervised Feature Learning Workshop* Vol. 5 (2011)

[36] Arunachalam, S., Gheorghiu, V., Jochym-O'Connor, T., Mosca, M. & Srinivasan, P. V. On the robustness of bucket brigade quantum RAM. *New J. Phys.* **17**, 123010 (2015)

[37] Li, Z., Liu, X., Xu, N. & Du, J. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett.* **114**, 140504 (2015)

[38] Chatterjee, R. & Yu, T. Generalized coherent states, reproducing kernels, and quantum support vector machines. Preprint at https://arxiv.org/abs/1612.03713 (2016)

[39] Wossnig, L., Zhao, Z. & Prakash, A. A quantum linear system algorithm for dense matrices. Preprint at https://arxiv.org/abs/1704.06174 (2017)

[40] Lau, H.-K., Pooser, R., Siopsis, G. & Weedbrook, C. Quantum machine learning over infinite dimensions. *Phys. Rev. Lett.* **118**, 080501 (2017)

[41] Clader, B. D., Jacobs, B. C. & Sprouse, C. R. Preconditioned quantum linear system algorithm. *Phys. Rev. Lett.* **110**, 250504 (2013)

[42] Sentís, G., Bagan, E., Calsamiglia, J., Chiribella, G. & Muñoz Tapia, R. Quantum change point. *Phys. Rev. Lett.* **117**, 150502 (2016)

[43] Dunjko, V., Friis, N. & Briegel, H. J. Quantum-enhanced deliberation of learning agents using trapped ions. *New J. Phys.* **17**, 023006 (2015)

[44] Paparo, G. D., Dunjko, V., Makmal, A., Martin-Delgado, M. A. & Briegel, H. J. Quantum speedup for active learning agents. *Phys. Rev. X* **4**, 031002 (2014)

[45] Sentís, G., Gut¸a˘, M. & Adesso, G. Quantum learning of coherent states. *EPJ Quant. Technol.* **2**, 17 (2015)

[46] Sentís, G., Calsamiglia, J., Muñoz-Tapia, R. & Bagan, E. Quantum learning without quantum memory. *Sci. Rep.* **2**, 708 (2012)

[47] Bisio, A., D'Ariano, G. M., Perinotti, P. & Sedlák, M. Quantum learning algorithms for quantum measurements. *Phys. Lett. A* **375**, 3425–3434 (2011)

[48] Sasaki, M., Carlini, A. & Jozsa, R. Quantum template matching. *Phys. Rev. A* **64**, 022317 (2001)

[49] Schuld, M., Sinayskiy, I. & Petruccione, F. Prediction by linear regression on a quantum computer. *Phys. Rev. A* **94**, 022342 (2016)

[50] Lloyd, S., Garnerone, S. & Zanardi, P. Quantum algorithms for topological and geometric analysis of data. *Nat. Commun.* **7**, 10138 (2016)

[51] Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014)