IRSTI 27.35.57 https://doi.org/10.26577/jpcsit20253105

Aksultan Mukhanbet[1,2], Nurtugan Azatbekuly[1,2*], Beimbet Daribayev[1,2]

[1] LLP DigitAlem, Almaty, Kazakhstan
[2] Al-Farabi Kazakh National University, Almaty, Kazakhstan
*e-mail: nurtugang17@gmail.com

DEVELOPMENT OF HYBRID QUANTUM-CLASSICAL MODELS FOR COMPUTER VISION

Abstract. This research explores the integration of quantum computing with classical machine learning to enhance data classification tasks using Quantum Neural Networks (QNN) and Parameterized Quantum Circuits (PQC). The hybrid approach leverages the advantages of both quantum and classical systems to improve the efficiency and accuracy of data processing. In this model, data is encoded into qubits using amplitude encoding, representing input vectors as amplitudes of quantum states. The QNN is initialized by placing the qubits in superposition using Hadamard gates, followed by data encoding with parameterized rotational gates that map classical data to quantum states using rotation angles. PQC plays a central role by applying layers of parameterized quantum operations to process data in the quantum space. These parameters are optimized during the training process, where a quadratic loss function minimizes the error between the predicted quantum states and the true class labels using gradient descent. Experiments conducted on the MNIST dataset show that the hybrid quantum-classical neural network (QCNN) with PQC achieves a classification accuracy of over 95%, highlighting its potential in machine learning applications. The results demonstrate that integrating quantum computing with classical machine learning enhances performance in complex data analysis tasks due to the exponential growth of quantum state space and the parallelism of quantum systems, making hybrid models promising for computer vision and classification tasks.

Key words: quantum neural networks, hybrid computing, data classification, machine learning, gradient descent.

1. Introduction

The rapid progress in the field of machine learning (ML) has significantly transformed various industries, including healthcare, finance, and consumer behavior analysis. Esteva et al. demonstrated that deep neural networks (DNNs) can classify skin cancer with accuracy comparable to that of experienced dermatologists, highlighting the potential of ML in medical diagnostics [1]. Furthermore, Ngai et al. analyzed the application of data mining techniques to detect financial fraud, creating a classification system that emphasizes the importance of data mining for identifying and preventing fraudulent activities [2]. Zhang et al. further contributed to this field by applying machine learning algorithms to predict consumer preferences and uncover gender biases in online reviews [3]. As the size and complexity of datasets increase, the limitations of traditional computational methods become increasingly apparent. Amodei et al. discuss the challenges associated with the exponential growth of data volume and the need for more efficient computational power [4]. This situation has led to a resurgence of interest in quantum computing, which promises to overcome these limitations by leveraging quantum mechanics to perform computations at unprecedented speeds. Waldrop describes the forthcoming challenges to Moore's Law, emphasizing the relevance of innovative computational paradigms [5].

Quantum algorithms, such as those proposed by Shor for factoring and discrete logarithms [6] and Grover for database searching [7], demonstrate significant advantages of quantum computing over classical methods. Biamonte et al. have since become pioneers in the integration of quantum computing with machine learning, coining the term "quantum machine learning" (QML), aimed at harnessing quantum mechanics to enhance learning capabilities [8]. This integration has sparked considerable research into quantum neural networks (QNNs), which utilize quantum circuits to model neural networks [9].

Various approaches to QML have emerged, including quantum support vector methods [11], which enhance the classification of large datasets, and quantum perceptron models [13]. Moreover, quantum data fitting algorithms and supervised learning demonstrate the potential of quantum systems to revolutionize traditional ML tasks [10][12]. The development of parameterized quantum circuits as models for machine learning further illustrates this trend [16].

Despite these achievements, the search for effective QNN architectures continues. Farhi and Neven explore the classification capabilities of quantum neural networks on near-term processors [20]. Meanwhile, Dunjko and colleagues highlighted the progress made in quantum reinforcement learning, suggesting that QML can expand its capabilities to tackle more complex decision-making problems [21]. Additionally, advancements in circuit-centric quantum classifiers illustrate the evolving landscape of QML, where quantum circuits are optimized to address specific machine learning tasks [18]. However, despite these advancements, many existing QML methodologies face limitations, particularly regarding scalability, sensitivity to noise, and the lack of reliable architectures suitable for various real-world applications.

In this research, we propose to develop a hybrid quantum-classical architecture specifically designed for computer vision applications. Our approach combines QNN with Parameterized Quantum Circuits (PQC) to enhance the efficiency of classifying image datasets such as MNIST. The hybrid model leverages the advantages of both quantum and classical systems, where quantum circuits are used for efficient data encoding and feature extraction, while classical networks handle the remaining computations. This integration allows us to utilize the exponential state space and parallelism of quantum systems, which could potentially lead to significant improvements in accuracy and computational efficiency.

2. Materials and Methods

In this section, we present our proposed combined classical and quantum computational methods. Specifically, we partially transform a classical neural network into a quantum neural network to create a hybrid quantum-classical neural network. The proposed methodology consists of three main components: data collection, model development and evaluation.

2.1. Data Collection and Encoding

The primary dataset used in this research is the MNIST dataset, a well-known benchmark in the field of machine learning, particularly for handwritten digit recognition tasks. The MNIST dataset consists of 70,000 grayscale images of handwritten digits, each measuring 28x28 pixels. This dataset is particularly valuable due to its simplicity and the extensive research that has been conducted using it, making it a standard for evaluating various machine learning algorithms.

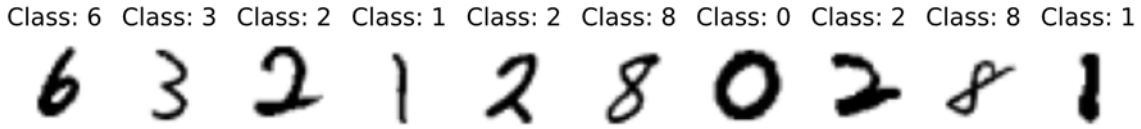

Figure 1 – MNIST Dataset

The MNIST dataset is publicly available and can be easily accessed through various libraries, including TensorFlow and PyTorch. These libraries provide straightforward interfaces for loading the dataset directly into the working environment, facilitating quick integration into the model development process. The dataset is divided into two parts: 60,000 training images and 10,000 testing images. This division ensures that the model can be trained on different samples and then evaluated based on unseen data to assess its generalization capabilities.

2.1.1 Data Preprocessing

Before using the MNIST dataset to train the hybrid model, several preprocessing steps need to be performed for effective data preparation:

Normalization: Pixel values of the images, which range from 0 to 255, will be normalized to values between 0 and 1. This normalization process

helps to improve convergence speed during model training and ensures that the model processes all input features equally.

Reshaping: The images will be reshaped according to the requirements of the neural network. For the CNN (Convolutional Neural Network) component, the images will be transformed into a 4D tensor with dimensions (batch_size, height, width, channels). In the case of grayscale images from MNIST, the channel dimension will be 1.

Data Augmentation: To increase the diversity of the training data and enhance the model's robustness, various data augmentation techniques will be applied. These methods may include random rotations, translations, scaling, and horizontal flips, allowing the model to learn to recognize digits from different perspectives.

2.1.2 Quantum Data Encoding

For the quantum component of the hybrid model, it is crucial to encode the preprocessed data into a quantum format that can be utilized by quantum circuits. Several methods exist for this purpose, and the choice of encoding can significantly impact the model's performance. In this study, we will consider amplitude encoding for quantum data encoding.

Amplitude encoding is a powerful method for representing classical data in a quantum state, allowing for efficient processing of high-dimensional data. In the context of the MNIST dataset, which consists of 28x28 grayscale images of handwritten digits, amplitude encoding enables us to encode each image as a quantum state using a logarithmic number of qubits relative to the number of pixels.

Given a normalized N-dimensional input vector $\vec{x}$ representing an image from the MNIST dataset, where N = 784 (the total number of pixels in a 28x28 image), we can encode the vector into the quantum state $|\phi(\vec{x})\rangle$ as follows:

$$|\phi(\vec{x})\rangle = \sum_{i=1}^{N} x_i |i\rangle \quad (1)$$

Here, x_i is the amplitude corresponding to the i-th pixel of the image, and $|i\rangle$ represents the i-th computational basis state. Each element x_i is derived from the pixel values of the image after normalization, ensuring that the total amplitude satisfies the normalization constraint $\| \vec{x} \| = 1$.

The main advantage of amplitude encoding lies in its ability to leverage the exponentially growing Hilbert space of quantum states, allowing us to represent high-dimensional data using a relatively small number of qubits. This feature is particularly useful when dealing with complex datasets, such as MNIST, which can exhibit significant variations in handwritten digits.

The preprocessing stage, which may include normalization, resizing, and noise suppression, lays the groundwork for subsequent stages by enhancing the quality of the input data.

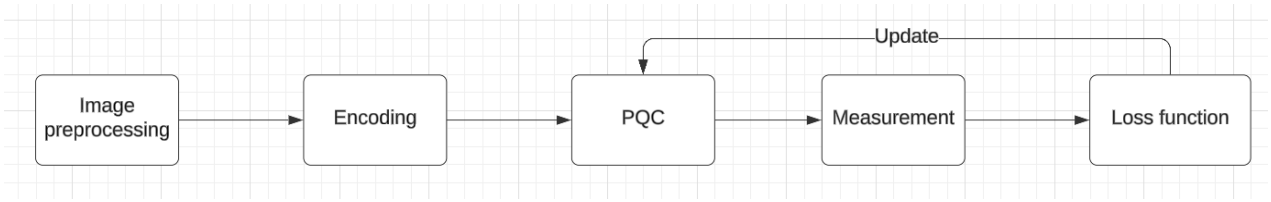

Figure 2 – Diagram of the hybrid model for image preprocessing

After preprocessing, classical information is transformed into quantum using methods such as amplitude encoding, where pixel values are represented as amplitudes of a quantum state. For example, the normalized input vector obtained from pixel data can be converted into a quantum state $|\phi(x)\rangle$, allowing the quantum system to process the data efficiently. The next step involves developing a PQC, which consists of parameterized quantum gates arranged in a manner that best suits the specific task. The analysis can be repeated multiple times, with the number of repetitions adjusted based on empirical results. After applying the analysis, the quantum state is measured, resulting in classical outcomes that are evaluated using a predefined loss function to assess the model's effectiveness. The cycle of measurements, feedback, and parameter adjustments continues until

the model converges and satisfactory performance in recognizing digits from the dataset is achieved. Overall, this integration of classical preprocessing with quantum computing methods allows researchers to leverage the unique capabilities of quantum neural networks to enhance the performance of machine learning algorithms in solving complex tasks such as digit recognition.

2.2. Quantum Circuit (Parameterized Circuit)

The QuantumMNISTRecognizer algorithm describes the steps for implementing a quantum circuit module designed for digit recognition in the MNIST dataset. The algorithm consists of three main procedures: INITIALIZE, TRAIN, and TEST, each serving a specific purpose in the quantum machine learning pipeline.

Algorithm 1: Quantum circuit module

```
procedure INITIALIZE(kernel_size, backend, shots)
 self.n_qubits ← kernel_size ** 2
 self._circuit ← Create a quantum circuit with n_qubits
 self.theta ← Create a list of Parameter objects for each qubit
 for i from 0 to n_qubits – 1 do
 self._circuit.rx(self.theta[i], i) // Apply RX gates to all qubits
 end for
 self._circuit.barrier() // Add a barrier
 self._circuit += random_circuit(self.n_qubits, 2)
 self._circuit.measure_all() // Measure all qubits
 Set the number of shots for execution
end procedure
procedure TRAIN(X_train, y_train, epochs)
 for each epoch in epochs do
 for each (x, y) in (X_train, y_train) do
 θ ← Learnable parameters for the current image x
        self._circuit.assign_parameters(θ) // Assign current parameters to the circuit
        TC ← Transpile the circuit for the specified backend
        Results ← Execute the circuit TC on the backend
        loss ← Compute the loss function using Results and y
        Update θ using an optimizer (e.g., gradient descent)
 end for
 end for
end procedure
procedure TEST(X_test)
 Predictions ← []
 for each x in X_test do
 θ ← Optimal parameters for x
 self._circuit.assign_parameters(θ) // Assign optimal parameters for testing
 TC ← Transpile the circuit for backend
 Results ← Execute the circuit TC on the backend
 p ← Measurement probabilities for each possible state
 pred ← Classification based on p
 Predictions.append(pred)
 end for
 return Predictions
end procedure
```

The INITIALIZE procedure begins by determining the number of qubits required for the quantum circuit, which is calculated as the square of the kernel_size parameter. This approach ensures that each pixel of the input image can be represented by a qubit, allowing the circuit to efficiently process image data. A quantum circuit is then created with the computed number of qubits, and a list of parameterized angles, called theta, is created for each qubit. These angles are specifically designed for use with RX gates that will be applied to the qubits. This initialization process lays a solid foundation for the quantum circuit, preparing it for effective training and testing in the digit recognition task.

The TRAIN procedure begins with iterating over the training data for a specified number of epochs, which determines how many times the entire dataset is processed. For each epoch, the algorithm goes through individual image-label pairs from the training set. At each step, the trainable parameters theta, which correspond to the rotation angles in the RX gates of the quantum circuit, are updated for the current input image. The updated values are then assigned to the circuit's parameters, adjusting the quantum circuit to process the image. Using these results, the loss function is computed by comparing the circuit's outputs with the actual label for the image, quantitatively determining how well the circuit performs the classification task. Finally, the trainable parameters are updated using an optimization method, such as gradient descent, to improve the circuit's performance in the next iteration. This process is repeated for each data point and continues for the specified number of epochs, gradually enhancing the model's ability to classify digits.

The TEST procedure focuses on evaluating the performance of the quantum model on a separate test dataset. The process begins by initializing an empty list to store predictions. For each test image, the optimal theta parameters, which were learned during the training phase, are assigned to the quantum circuit. As in the training procedure, the circuit is then transpiled and executed on a quantum backend. After execution, the measurement probabilities for different quantum states are collected. These probabilities are analyzed to determine the most likely class or label for the image, forming the predicted classification. The prediction is then added to the list of results. This process is repeated for all images in the test dataset. Once all predictions have been made, the list is returned, providing the final output of the model's performance on unseen data. This testing phase is crucial for assessing the generalization capability of the quantum model and ensuring its ability to accurately classify new images beyond the training set.

2.3. Classical Neural Network Architecture

To address the image classification task using the MNIST dataset, a simple classical neural network architecture can be utilized. This architecture consists of three main layers: an input layer, one or more hidden layers, and an output layer. The input layer contains 784 neurons, corresponding to each pixel in the 28x28 image, where the pixel values are normalized within the range of 0 to 1.

The hidden layers can be implemented using fully connected (dense) neurons. For example, a single hidden layer with 128 neurons using the ReLU (Rectified Linear Unit) activation function can be effective for feature extraction from the images. After processing in the hidden layers, the output layer consists of 10 neurons, each corresponding to one of the ten digits (from 0 to 9). The softmax activation function is used in the output layer, providing a probability for each class, which allows for easy interpretation of the classification results.

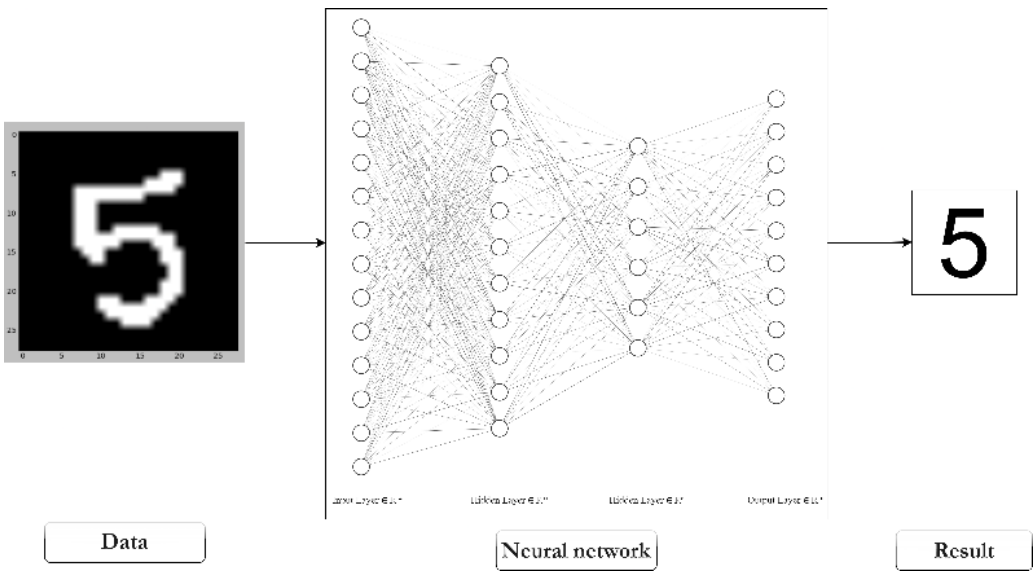

Figure 3 – Classical Architecture with CNN for the MNIST Dataset

This architecture is basic and can be further improved using regularization methods, such as dropout, to reduce the likelihood of overfitting. To optimize the model training process, loss functions such as categorical cross-entropy were also employed.

2.4. Hybrid Quantum Neural Network Architecture

On the other hand, a hybrid quantum neural network architecture that combines classical and quantum approaches represents a more complex and innovative approach to classifying images from MNIST. In this architecture, the initial part of the model can remain classical, where images first pass through several fully connected layers for feature extraction, similar to the simple architecture.

However, after feature extraction, the classical part of the network will interact with the quantum part. For this, PQC can be used, which accept data in the form of amplitudes obtained from the previous stage. This quantum part of the network is responsible for complex nonlinear transformations, which can lead to improved classification quality, especially in tasks where traditional approaches have limitations.

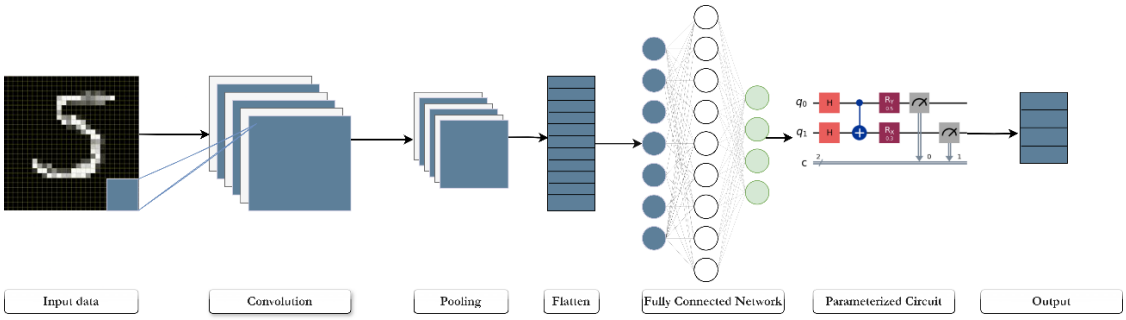

Figure 4 – Architecture of a Hybrid Quantum-Classical Neural Network

As the output layer, a classical structure with 10 neurons and a softmax activation function can be used again to obtain class probabilities. This hybrid approach leverages the strengths of both classical and quantum methods, providing a deeper representation of the data and potentially improving classification accuracy on MNIST compared to purely classical architectures.

2.5. Combining Classical and Quantum Results

Decision Making: The final classification of actions is determined by combining the results from both the CNN and QNN components. This can be achieved using methods like weighted averaging or summation, where the predictions from each model contribute to the final decision.

To integrate quantum and classical neural networks into our hybrid network, we include a hidden layer known as the PQC (discussed in section 2.2). The circuit uses the classical input vector to set the rotation angles of quantum gates. The outputs from the previous layer of the neural network are fed as inputs into this parameterized circuit. The measurement statistics from the parameterized circuit are collected and used as input for the subsequent neural network layer. This process continues until the final output layer is reached.

3. Results and Discussion

In this section, we present the experimental results of our proposed the hybrid quantum-classical neural network (QCNN) model. The experiments were conducted on a computer equipped with an Nvidia RTX 4080 GPU. Despite the availability of high-end quantum computing platforms from companies such as IBM and Google, these services are currently not accessible in certain regions. This limitation significantly hampers the ability of researchers and developers from CIS countries, including Kazakhstan, to leverage cutting-edge quantum technologies for their projects.

Nevertheless, various strategies can help mitigate these access barriers and will be explored in future studies. Partnerships and academic agreements can often grant subsidized or specialized access, even in areas with regulatory hurdles. Additionally, cloud quantum services from providers like Amazon Braket and Microsoft Azure Quantum represent promising alternatives. In future studies, these approaches expand the scope of cutting-edge experiments and promote inclusive growth in the quantum research community despite existing geographic limitations.

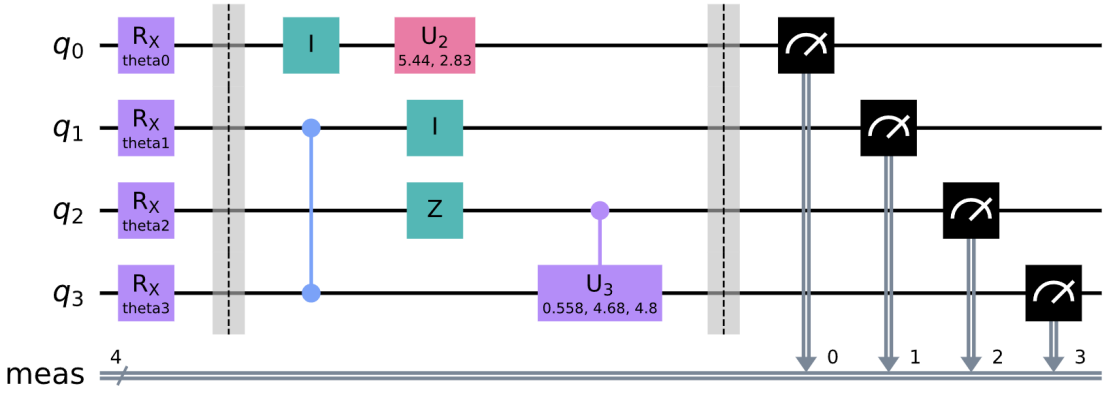

Figure 5 – Results of the quantum circuit for image classification in the hybrid QCNN

In the circuit, the theta angles are specifically designed for use with RX gates, which are applied to the qubits. The procedure is repeated for each qubit by applying an RX gate parameterized with the corresponding angle, effectively preparing the qubits for subsequent quantum operations. After applying these gates, a barrier is introduced, which marks the end of the initial circuit preparation phase. After crossing the barrier, a random circuit is added, creating additional entanglement between the qubits, enhancing the circuit's ability to capture complex relationships in the data. Finally, the circuit measures all the qubits, enabling result extraction after execution, and the number of repetitions for quantum execution is set, determining how many times the circuit will be run to gather statistics from the measurement results.

In the case of the MNIST dataset, comparing the performance of a classical CNN and the hybrid model reveals important insights into how well each model handles digit recognition tasks. For the classical CNN, the training accuracy is 0.92, indicating that the model successfully learned from the training data. This accuracy is consistent with the test accuracy of 0.92, showing that the model generalizes well to unseen data. The close alignment of these accuracy values suggests that the CNN is not overfitting and has effectively learned the features of the MNIST dataset.

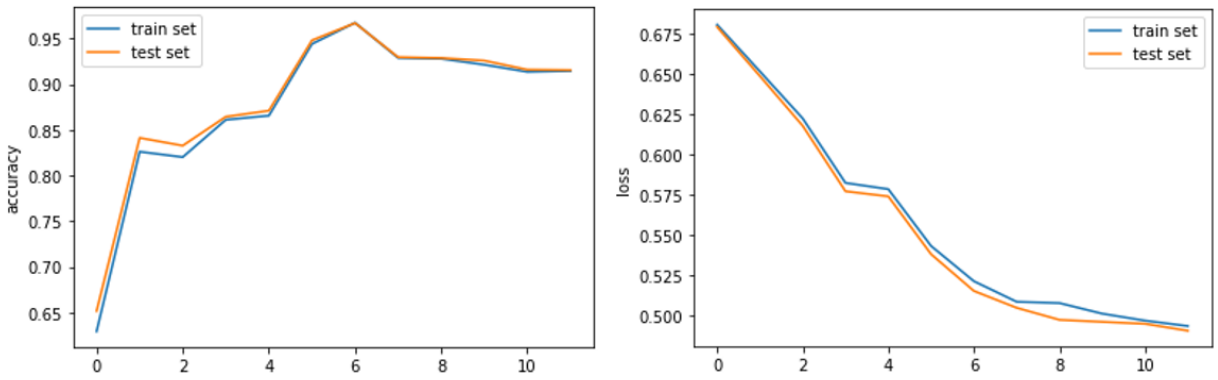

Figure 6 – Accuracy and Loss During Neural Network Training and Testing

Regarding losses, the CNN shows a training loss of 0.5 and slightly lower test loss of 0.4. This reduction in loss between the training and testing phases reflects improved model performance when applied to unseen data. The low loss values also confirm that the CNN is effectively optimizing, minimizing the error between predicted and actual digit labels.

For the MNIST dataset, the performance of the hybrid QCNN shows improvements over the classical CNN in both accuracy and generalization. During training, the hybrid QCNN achieves an accuracy of 0.95, higher than the classical CNN's 0.92, indicating that the hybrid model is learning the dataset features more effectively. Furthermore, the test accuracy of 0.96 demonstrates that the model generalizes even better to unseen data compared to the CNN, where test accuracy remained at 0.92.

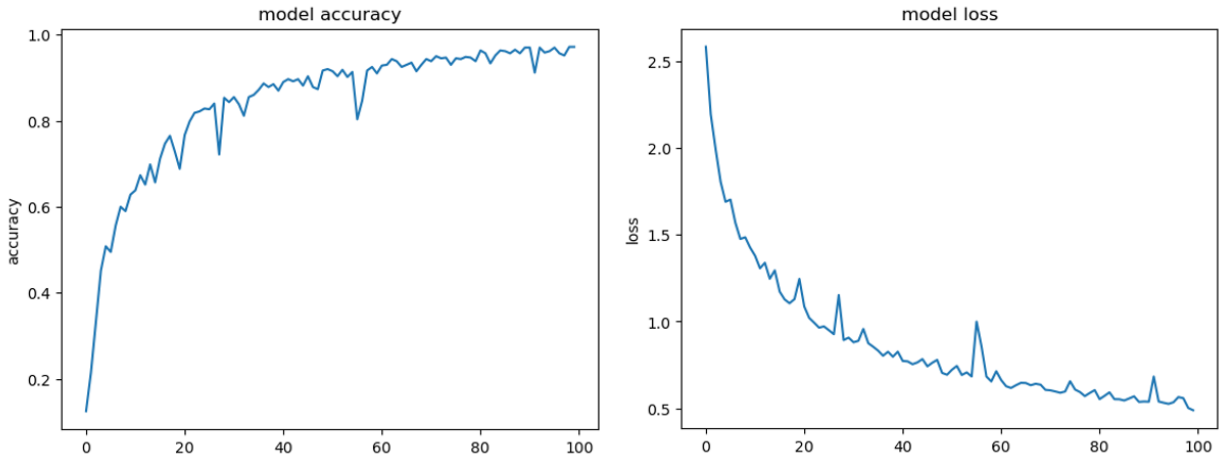

Figure 7 – Loss and Accuracy Results for the hybrid QCNN

However, the loss values for the hybrid QCNN are slightly higher than those of the classical CNN. The training loss stands at 0.6, while the test loss decreases to 0.5, reflecting a general reduction in error as the model is tested on new data. This indicates that while the hybrid QCNN achieves higher accuracy, further optimization or parameter tuning may be needed to reduce training losses. Nonetheless, the lower test losses show that the model performs better with unseen data, effectively capturing the complexity of the MNIST dataset compared to the classical approach.

The image displays the results of the hybrid QCNN model trained to recognize handwritten digits from the MNIST dataset. The model's predictions for a sample set of digits are as follows: 4, 2, 4, 6, 8, 1, 5, and 7. These predictions match the corresponding images, demonstrating that the hybrid QCNN successfully recognizes the handwritten digits.

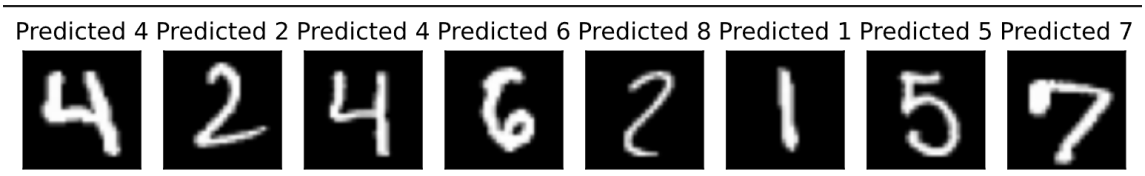

Figure 8 – MNIST Classification Results Using Hybrid QNN

This result highlights the model's effectiveness, particularly in terms of generalizing across various handwriting styles, which is a critical challenge in the MNIST dataset. Given that this is a hybrid quantum part of the model, it underscores the potential of combining quantum neural networks with classical methods to improve accuracy in image recognition tasks, such as digit classification.

The comparison between the classical CNN and the hybrid QCNN for the MNIST dataset shows significant differences in performance, as illustrated in Table 1.

Table 1 – Comparison of Classical CNN and Hybrid QCNN for MNIST Dataset

Computation type	train	test
CNN accuracy	0.92	0.92
CNN loss	0.5	0.4
Hybrid QCNN accuracy	0.95	0.96
Hybrid QCNN loss	0.6	0.5

The CNN achieved an accuracy of 0.92 on both the training and test datasets, with corresponding losses of 0.5 during training and 0.4 during testing. This indicates that the model generalizes well to both sets, maintaining stable performance without overfitting or underfitting.

In contrast, the hybrid QCNN outperformed the classical CNN in terms of accuracy, reaching 0.95 on the training set and 0.96 on the test set. However, the loss values were higher compared to the CNN: 0.6 for training and 0.5 for testing. While the hybrid QCNN demonstrates better accuracy, the increased losses suggest that it may converge more slowly or may require further tuning, especially regarding parameters and entanglement layers in the quantum part of the model.

The comparison of execution times for classical and quantum approaches highlights different performance characteristics for both types of computations. The classical CNN running on a CPU took 30.065432 seconds, while the same classical CNN executed on an Nvidia RTX 4070 GPU demonstrated significant acceleration, completing in just 12.012345 seconds. This underscores the substantial advantage of using GPUs for parallel operations typical of CNNs, where GPU architecture excels in handling large-scale matrix multiplications and convolutions.

Table 2 – Comparison of Model Execution Time on Different Devices

Computation type	Device	Time (seconds)
Classical CNN	CPU	30.065432
Classical CNN	Nvidia RTX 4070	12.012345
Hybrid QCNN	QASM Simulator	36.123456
Hybrid QCNN	Statevector Simulator	28.038765

Conversely, modeling the hybrid QCNN shows a different performance profile. The QASM simulator, which models quantum circuits with measurement processes, took **36.123456 seconds**, reflecting the computational complexity associated with simulating quantum operations on classical hardware. However, the statevector simulator, which only simulates the evolution of quantum states without measurement, was faster, completing its task in **28.038765 seconds**. This result indicates that the statevector simulator can handle quantum operations more efficiently than the QASM simulator, but both are still slower compared to the classical CNN running on a GPU.

Losses when using quantum models can arise for several reasons. One of the key issues is quantum noise and errors that accumulate due to imperfections in quantum gates and measurements, which

is particularly critical for complex quantum algorithms requiring numerous operations. Additionally, the limited coherence of qubits–the time during which they can maintain their quantum state–means that longer operations are susceptible to decoherence. Due to these factors, some quantum models become challenging to work with and less efficient than expected.

Overall, while quantum simulators bring the potential for new computational paradigms, classical GPUs remain significantly better in terms of pure speed for CNN tasks. As quantum hardware evolves, this performance gap may narrow, but for now, quantum simulation remains a more computationally intensive process compared to highly optimized classical computations on modern GPUs. Moreover, the hybrid QCNN demonstrates superior predictive accuracy, highlighting the potential benefits of quantum computing in neural network architectures, although it may still face optimization and convergence challenges, as evidenced by higher loss values compared to CNN.

5. Conclusions

In conclusion, this study examined the integration of hybrid quantum-classical neural networks into image recognition tasks using the MNIST dataset, providing a comprehensive analysis of their performance compared to traditional CNNs. The primary objective was to assess the advantages and limitations of quantum neural networks in practical machine learning tasks, particularly in digit classification, which serves as a benchmark problem in the field of computer vision.

The classical CNN demonstrated reliable and stable performance, achieving an accuracy of 92% on both training and test datasets, accompanied by relatively low loss values. These results align with the expected outcomes from a well-tuned classical model, showcasing its ability to generalize across different datasets without significant performance degradation. In contrast, the hybrid QCNN exhibited a promising leap in accuracy, surpassing the CNN with a training accuracy of 95% and a testing accuracy of 96%. These findings indicate that the integration of quantum computing, even in a hybrid form, can substantially enhance the learning and generalization capabilities of neural networks. The quantum component of the hybrid QCNN introduces unique data processing mechanisms, leveraging properties such as superposition and entanglement, which allow the model to explore broader and more complex solution spaces and potentially uncover deeper patterns in the data. This advantage was particularly evident in the higher testing accuracy of the model, indicating its ability to generalize beyond the training set.

However, despite the higher accuracy, the hybrid QCNN exhibited higher loss values during both training and testing, indicating a potential trade-off between accuracy and model optimization. This increase in loss suggests that, while the hybrid QCNN system is effective at yielding accurate predictions, its optimization may be more challenging due to the additional complexity introduced by the quantum level. The issues of convergence and parameter tuning in quantum circuits represent areas that require further refinement. Higher losses may be associated with entanglement layers or the design of quantum gates, which might necessitate more sophisticated optimization methods or deeper parameterized quantum circuits for smoother learning dynamics.

Funding

This study was funded by Grant No. AP19576314 “Development of high-performance hybrid algorithms for solving oil displacement problems based on classical (GPU) and quantum computing” of the Ministry of Science and Higher Education of the Republic of Kazakhstan.

Author Contributions

Conceptualization, A.M.; Methodology, A.M.; Software, A.M.; Validation, A.M. and B.D.; Formal Analysis, A.M.; Investigation, A.M.; Resources, A.M.; Data Curation, A.M.; Writing – Original Draft Preparation, A.M.; Writing – Review & Editing, N.A.; Visualization, N.A..; Supervision, B.D.; Project Administration, B.D.; Funding Acquisition, B.D.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Esteva et al., "Dermatologist-level classification of skin cancer with deep neural networks," Nature, vol. 542, no. 7639, pp. 115–118, 2017. [Online]. Available: https://doi.org/10.1038/nature21056

2. E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," Decision Support Systems, vol. 50, no. 3, pp. 559–569, 2011. [Online]. Available: https://doi.org/10.1016/j.dss.2010.08.006

3. D. Zhang, L. Ma, X. Huang, and S. Wang, "Predicting consumer preferences and detecting gender bias in online reviews using machine learning algorithms," Computers in Human Behavior, vol. 98, pp. 120–134, 2019. [Online]. Available: https://doi.org/10.1016/j.chb.2019.04.011

4. D. Amodei and D. Hernandez, "AI and compute," OpenAI Blog, 2018. [Online]. Available: https://openai.com/blog/ai-and-compute/

5. M. M. Waldrop, "The chips are down for Moore's law," Nature News, vol. 530, no. 7589, pp. 144–147, 2016. [Online]. Available: https://doi.org/10.1038/530144a

6. P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in Proceedings of the 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124–134. [Online]. Available: https://doi.org/10.1109/SFCS.1994.365700

7. L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

8. J. Biamonte et al., "Quantum machine learning," Nature, vol. 549, no. 7671, pp. 195–202, 2017. [Online]. Available: https://doi.org/10.1038/nature23474

9. M. Schuld, I. Sinayskiy, and F. Petruccione, "The quest for a quantum neural network," Quantum Information Processing, vol. 13, no. 11, pp. 2567–2586, 2014. [Online]. Available: https://doi.org/10.1007/s11128-014-0809-8

10. N. Wiebe, D. Braun, and S. Lloyd, "Quantum algorithm for data fitting," Physical Review Letters, vol. 109, no. 5, p. 050505, 2012. [Online]. Available: https://doi.org/10.1103/PhysRevLett.109.050505

11. P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," Physical Review Letters, vol. 113, no. 13, p. 130503, 2014. [Online]. Available: https://doi.org/10.1103/PhysRevLett.113.130503

12. S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," arXiv preprint arXiv:1307.0411, 2013. [Online]. Available: https://arxiv.org/abs/1307.0411

13. N. Wiebe, A. Kapoor, and K. M. Svore, "Quantum perceptron models," arXiv preprint arXiv:1602.04799, 2016. [Online]. Available: https://arxiv.org/abs/1602.04799

14. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/

15. L. Blum and R. L. Rivest, "Training a 3-node neural network is NP-complete," Neural Networks, vol. 5, no. 1, pp. 117–127, 1992. [Online]. Available: https://doi.org/10.1016/S0893-6080(05)80010-3

16. M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," Quantum Science and Technology, vol. 4, no. 4, p. 043001, 2019. [Online]. Available: https://doi.org/10.1088/2058-9565/ab4eb5

17. V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," Nature, vol. 567, no. 7747, pp. 209–212, 2019. [Online]. Available: https://doi.org/10.1038/s41586-019-0980-2

18. M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," Physical Review A, vol. 101, no. 3, p. 032308, 2020. [Online]. Available: https://doi.org/10.1103/PhysRevA.101.032

19. Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, "Quantum neuron: an elementary building block for machine learning on quantum computers," arXiv preprint arXiv:1711.11240, 2017. [Online]. Available: https://arxiv.org/abs/1711.11240

20. E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," arXiv preprint arXiv:1802.06002, 2018. [Online]. Available: https://arxiv.org/abs/1802.06002

21. V. Dunjko, J. M. Taylor, and H. J. Briegel, "Advances in quantum reinforcement learning," arXiv preprint arXiv:1709.02779, 2017. [Online]. Available: https://arxiv.org/abs/1709.02779

Information About Authors:

Aksultan Mukhanbet is a researcher at LLP DigitAlem and a PhD student in the Computer Science Department at Al-Farabi Kazakh National University (Almaty, Kazakhstan, mukhanbetaksultan0414@gmail.com). His research interests include quantum computing, quantum machine learning, quantum optimization, and computer vision. ORCID ID: 0000-0003-4699-0436.

Nurtugan Azatbekuly is a junior researcher at LLP DigitAlem and a master's student in the Computer Science Department at Al-Farabi Kazakh National University (Almaty, Kazakhstan, nurtugang17@gmail.com). His research interests focus on the analysis and development of computer vision algorithms and quantum computing. ORCID ID: 0009-0007-5843-8995.

Beimbet Daribayev, PhD, is a senior researcher at LLP DigitAlem and an Associate Professor in the Computer Science Department at Al-Farabi Kazakh National University (Almaty, Kazakhstan, beimbet.daribayev@gmail.com). His research interests include quantum computing, machine learning, and high-performance computing. ORCID ID: 0000-0003-1313-9004.