A Novel Quantum Convolution Neural Network for Image Classification Applications

Venkatachalapathy Madhavanna Venkatappa
School of Electronics and Communication Engineering, REVA University, Bengaluru, Karnataka, India
mvvcpathy@gmail.com (corresponding author)

Venkateshappa
School of Electronics and Communication Engineering, REVA University, Bengaluru, Karnataka, India
venkateshappa@reva.edu.in

Received: 30 May 2025 | Revised: 17 July 2025 | Accepted: 23 July 2025

 DOI: https://doi.org/10.48084/etasr.12416

ABSTRACT

Image classification plays a vital role in large-scale data analysis, especially in object recognition tasks using advanced Deep Learning (DL) frameworks. However, the growing complexity and computational demands of modern DL models have introduced challenges related to scalability and efficiency. Quantum Computing (QC) has emerged as a promising alternative, capable of addressing these limitations by leveraging the principles of Quantum Machine Learning (QML). However, many existing QML models require a large number of qubits, which poses limitations within the current Noisy Intermediate-Scale Quantum (NISQ) era. This work introduces a novel Adaptive Quantum Convolutional Neural Network (AQCNN) designed for efficient and scalable image classification within the constraints of NISQ devices. Addressing the limitations of existing QML approaches, particularly the high qubit requirements, AQCNN incorporates a resource-efficient quantum convolutional layer that performs localized quantum filtering using parameterized quantum circuits. A classical preprocessing layer encodes input features to reduce qubit load, followed by quantum embedding and hybrid quantum-classical layers that optimize feature extraction and classification performance. The model leverages an adaptive quantum convolution strategy, minimizing quantum gate depth and circuit complexity while preserving critical spatial hierarchies in image data. Evaluated on benchmark datasets, AQCNN achieved 95.88% accuracy on MNIST and 95.68% on FMNIST, outperforming comparable QML architectures. Additionally, the model supports scalable execution through parallel quantum circuit arrays, enabling practical deployment on current quantum hardware. This architecture demonstrates a significant advance in quantum-assisted image classification, balancing performance with qubit and gate efficiency. The integration of adaptive quantum convolution and hybrid processing not only enhances classification accuracy but also provides a viable path forward for deploying QML solutions under realistic hardware constraints.

Keywords-image classification; quantum computing; quantum machine learning; adaptive quantum convolutional neural network; FMNIST; MNIST; noisy intermediate-scale quantum

I. INTRODUCTION

Image processing and Computer-Vision (CV) applications have gained significant importance due to their cost-effectiveness and environmentally friendly assessment capabilities compared to traditional methods. Among these technologies, the use of Earth-observing satellites for image classification has emerged as a prominent area of research within CV, enabling the extraction of vital geographic and ecological information. However, with the exponential increase in data volume and the growing complexity of Machine-Learning (ML) algorithms, the computational demand for processing high-resolution satellite images has become a major obstacle to the widespread adoption of traditional ML techniques.

To address these challenges, researchers are exploring the potential of quantum technology, also known as Quantum Computing (QC), which is expected to handle computational tasks that are currently intractable for classical systems [1]. As a result, investigating the application of QC for complex image classification is a critical and promising area of study. Quantum Machine Learning (QML), which combines ML techniques with the power of QC, is expected to have transformative impacts across domains, including object classification and signal processing. QML has shown promise in several areas, such as data processing, segmentation, optimization, and the development of quantum correlation indicators [2]. Given this potential, QML has gained considerable attention in recent years.

The current generation of quantum processors, referred to as Noisy-Intermediate-Scale Quantum (NISQ) devices, consists of roughly 10-100 physical qubits [3]. Despite being inherently noisy, these devices are suitable for certain types of QML models, particularly for image classification tasks [1]. QC offers the potential to accelerate complex computations for image interpretation, producing classification outputs that align well with the probabilistic nature of QML methods.

Various QML approaches have been proposed using quantum annealers and Parametric Quantum Circuits (PQCs). A notable contribution was presented in [1], developing a Hybrid Quantum-Classical Convolutional Neural Network (CNN) (HQC-CNN) to efficiently extract features from Earth observation data. This study employed an encoding scheme to reduce quantum resource requirements and demonstrated high accuracy using the TensorFlow Quantum platform. Similarly, in [2], a QC-based classification framework used single-qubit encoding, called Single-Qubit-based Deep Quantum Neural-Network (SQ-DQNN), achieving classification accuracies of 82.5% (ORL Face), 89.5% (FMNIST), and 94.6% (MNIST). In [4], an image classification method used a Quantum Autoencoder (QAE) to enhance performance in noisy environments, achieving 92% accuracy on the Fashion-MNIST (FMNIST) dataset. In [5], the Resource-Effective Q-CNN (REQ-CNN) incorporated pooling and fully connected layers and achieved 92.59% accuracy on the FMNIST and MNIST datasets. Quantum Deep-CNN (QDCNN) [6] integrated quantum convolution layers and hybrid training, inspired by variational quantum techniques, achieving 91.40% accuracy on GTSRB and 98.97% on MNIST.

The Quantum Pseudo-Transposed CNN (QPTCNN) [7] used ring-structured circuits and rotational-angle encoding. QPTCNN had two variants, developed using CRY and CNOT gates, which achieved accuracies of 83.5% and 82.0% on FMNIST, and 91.0% on MNIST. These advances underline the growing importance of QML in image classification. However, current NISQ machines lack sufficient qubits and fault tolerance to support real-world QML applications. Thus, application-specific quantum simulators are essential for validating the practical feasibility of QML. Despite limitations, studying QML remains critical for unlocking the potential of future quantum technologies. Moreover, most existing satellite image classification research relies heavily on DL techniques, particularly CNNs, due to their adaptability and feature extraction efficiency [4, 5]. Thus, this work proposes a novel Q-CNN model, named Adaptive Quantum Convolutional Neural Network (AQCNN), inspired by [6]. AQCNN uses parametric quantum circuits and amplitude encoding for efficient feature extraction with fewer qubits. The classification process is parallelized using a hybrid processing array, enabling better scalability and power efficiency compared to Graphic Processing Unit (GPU) or Field-Programmable Gate-Array FPGA-based arrays. The contributions of this research work are as follows:

- Introduces a novel quantum neural network that adapts effectively to diverse classification tasks.
- A parallel processing array uses a hybrid TENSORGPU (TGPU) platform to accelerate the execution of AQCNN.
- Incorporates quantum convolution, which refers to the application of localized, parameterized quantum circuits that act as filters over input data, mimicking classical convolutional operations while exploiting quantum parallelism to capture complex feature correlations with reduced resource usage.
- Experiments using MNIST and FMNIST classification tasks show that AQCNN achieves higher accuracy than current state-of-the-art methods.

II. METHODOLOGY

This section introduces the AQCNN architecture designed for image classification. The model integrates a hybrid quantum-classical framework with an adaptive quantum convolution layer to efficiently extract features. A systolic array-based task processing model is proposed to enhance execution efficiency, enabling high-accuracy inference with reduced computational time and power consumption.

A. AQCNN Design

Traditional CNNs usually have alternating pooling and convolutional layers. In traditional CNNs, the convolution layer helps determine the patterns and input characteristics. Further, the pooling layer in the CNN helps reduce the dimensions of the output results of the convolution layer by decreasing computation overhead. The Q-CNN is similar to CNN architecture but has alternating pooling and convolution layers. The convolution layer of Q-CNN captures better inherent characteristics from the image compared to the CNN approach because of quantum entanglement and quantum parallelism techniques.

Using the concept of Q-CNN, this work presents the AQCNN architecture for image classification (Figure 1). AQCNN builds on traditional Q-CNN hidden-layer designs to leverage quantum advantages in feature extraction and dimensionality reduction. During the data analysis phase, the image data undergoes preprocessing and is divided into training and testing sets. The AQCNN model is composed of an encoding layer, a hidden layer (including quantum convolution and pooling), and an output layer with an objective function for optimization. The encoding layer transforms classical image data into quantum states (qubits). A classical input layer is added before quantum encoding to overcome limitations such as the need for square image padding in prior Q-CNNs. The quantum embedding layer within this stage utilizes Adaptive Quantum Circuits (AQCs) (Figure 2). Furthermore, the hidden layer is responsible for extracting features using quantum convolution. A series of unitary operations captures qubit interactions. The quantum pooling layer reduces the dimensionality by selectively discarding qubits. Finally, the output layer transforms the final quantum state into classical output via measurement, using expectation values of Pauli operators (σ_α). The objective function is further evaluated using classical optimization, where the parameters are iteratively updated to improve model performance.

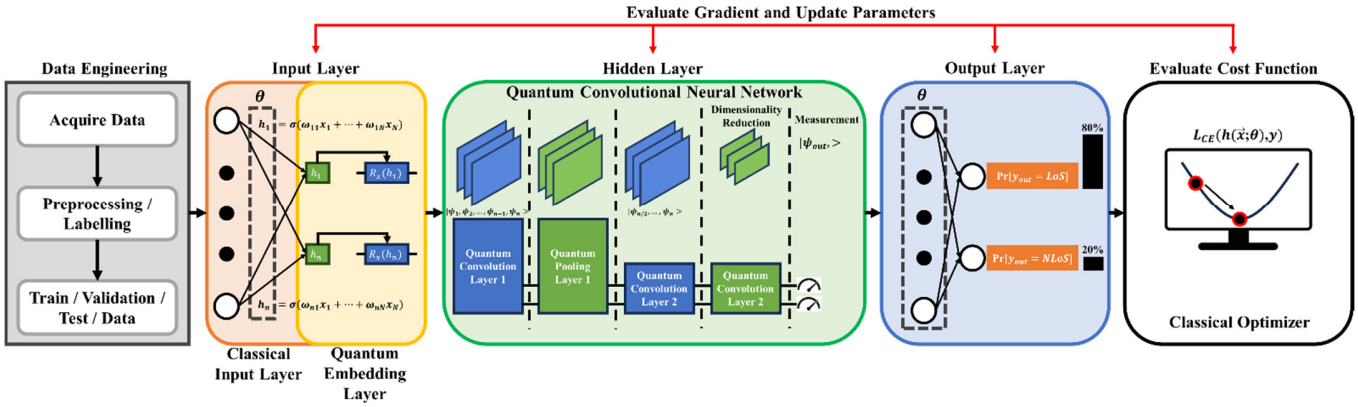

Fig. 1. AQCNN architecture for image classification.

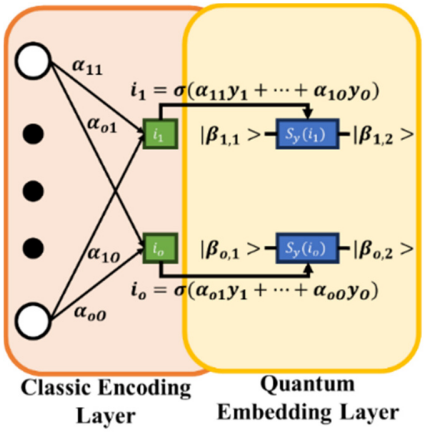

Fig. 2. The classical encoding layer with a quantum embedding layer.

In addition, in the pooling layer, the outcome achieved by the convolution layer by considering the qubits is decreased by performing different operations. In the pooling layer, some qubits are discarded because qubits that are idle or not effectively mapped to tasks may still require control and error correction operations, which leads to additional power usage and delays in overall system execution. Hence, this optimization is necessary because inactive or redundant qubits can increase energy consumption and execution time, as they continue to use system resources even when not performing useful computations. Moreover, the Q-CNN makes certain about non-linearities and provides better feature size reduction by decreasing the overall degree of freedom [8]. The measurement output is considered as the input for the output layer, represented as $|\beta_{output}\rangle$. The measurement output is attained by evaluating the expectation output from the final qubit utilizing the Pauli variable σ_a. In the output layer, the data is converted by extracting the output achieved by the final quantum pooling layer. The objective function is evaluated using classical optimization. The parameters are constantly updated as in the classical optimization process. Current Q-CNN approaches need extra padding or resizing of the input image, which leads to poor performance and higher computational costs. Instead, this work presents an input layer before the quantum layer to solve this problem, as shown in Figure 2. The advantage of AQCNN is that it uses flexible qubits and overcomes the representation limit within the output layer.

1) Layer 1-Encoding Layer

The encoding layer of the presented AQCNN comprises a classical input layer along with a quantum embedding layer. The complete flow is presented in Figure 2, where the quantum embedding layer consists of AQCs.

In the classical input-layer, the input image is taken as a vector in the form of $Y = [y_1, \ldots, y_O]^U$. In the AQCNN architecture, the vector is transformed into a rotation angle for AQC. Consider the output from the input activation layer as $[i_1, \ldots, i_o]^U$. Hence, the output i_o for the o^{th} rotation angle can be represented using (1), where $\sigma(\cdot)$ represents the activation function. From this, the initial states $|\beta_{o,1}\rangle$ for the o^{th} qubits can be represented using (2). The encoded quantum states achieved from (2) are further utilized as input for the Q-CNN layer (hidden layer).

$$i_o = \sigma\left(\alpha_{o|y|} + \cdots + \alpha_o O y_O\right) \quad (1)$$

$$|\beta_{o,2}\rangle = S_y(i_o)|\beta_{o,1}\rangle \quad (2)$$

2) AQCs to Q-CNN layer

In this layer, the patterns/features from qubits are extracted using the Quantum Convolution-Layer (QCL), which utilizes the random two-qubit unitary gates. The extraction process from AQCs to the QCL for the AQCNN architecture is detailed in [9, 10]. This work presents a novel AQC for building QCLs; thus, it can be considered as a benchmark for the upcoming AQC approaches. Moreover, to build an ideal circuit to compute $O(\tau, \varphi, \omega)$ to achieve a two-qubit AQC from V is the foundation of the benchmark AQC. In the $V \in \mathrm{V}$ unitary matrix, each unitary matrix is broken down as:

$$V = (B_1 \otimes B_2) \cdot O(\tau, \varphi, \omega) \cdot (B_3 \otimes B_4) \quad (3)$$

where $B_j \in \mathrm{V}$ is the unitary matrix, $\otimes$ represents the tensor-product, and $O(\tau, \varphi, \omega) = exp\left(j\left[\tau\sigma_y \otimes \sigma_y + \varphi\sigma_z \otimes \sigma_z + \omega\sigma_a \otimes \sigma_a\right]\right)$,with $\tau, \varphi, \omega \in S$ being the different parameters. The different parameters are set as follows: $\varphi = 2\phi - \frac{\pi}{2}$, $\tau = \frac{\pi}{2} - 2\theta$, and $\omega = \frac{\pi}{2} - 2\lambda$. Utilizing (3), the AQCs can be

understood using the operation of a unitary matrix $V \in \mathrm{V}$. Moreover, the AQCs are optimal according to [11]. Hence, the output of the l^{th} QCL is represented using:

$$|\beta_j(\theta_j)\rangle\langle\beta_j(\theta_j)| = \mathcal{T}_{H_j}\left(V_j(\theta_j)|\beta_{j-1}\rangle\langle\beta_{j-1}|V_j(\theta_j)|\right) \quad (4)$$

where $\mathcal{T}_{H_j}(\cdot)$ denotes the partial-trace operation for the system H_j, V_j denotes the AQC operation (QCL), and θ_j denotes AQC parameters for i^{th} QCL. In this work, some parameterized quantum circuits have been restricted because of the specific subspace of Hilbert space as the training process proceeds.

3) Quantum Circuit to Pooling-Layer

The main objective of this layer is to reduce the dimensions of the quantum circuit or the total qubits while keeping as much data as possible in comparison with the preceding QCL. However, there are no approaches that can be used to reduce the overall qubits in every Quantum Pooling Layer (QPL). Moreover, it is assumed that the QPL aims to reduce the dimensions from O qubits to $\frac{N}{2}$ qubits quantum circuit, thereby reducing the total qubits in every QPL. Hence, the QPL discards one qubit from every set of qubits. In the discarding process, the qubits are not evaluated using a quantum circuit. Moreover, the QPL merges the data of two qubits into a single qubit. Every two-qubit quantum circuit is functional for various sets of qubits for combining data from the preceding QCL. This work utilizes AQC for constructing QPL. The Quantum Pooling-Circuit (QPC) 1 comprises two controlled rotations, S_a and S_y [12]. In QPC 1, the targeted qubit is affected because of the control qubit state. To solve this, the QPC 2 $V_{\mathcal{P}}(\theta)$ is presented. The structure of QPC 2 is similar to that of QPC 1, with some modifications in its structure. The outcome of QPL 1 with QPC 2 can be defined using (5), where $V_{\mathcal{P}}$ represents AQC operations or the pooling layer and θ_j represents the parameters associated with the AQC of the j^{th} QPL.

$$|\beta_j(\theta_j)\rangle\langle\beta_j(\theta_j)| = \mathcal{T}_{H_j}\left(V_{\mathcal{P}}(\theta_j)|\beta_{j-1}\rangle\langle\beta_{j-1}|V_{\mathcal{P}}(\theta_j)|\right) \quad (5)$$

B. Systolic Array Task Processing Model

This section discusses the computational resources required to execute AQCNN on a hybrid TGPU architecture, which integrates Tensor Processing Units (TPUs) and GPUs. TPUs offer low-latency, optimized neural network execution, while GPUs provide high-throughput, parallel task processing. In a TGPU system, the strengths of both are combined to efficiently handle the diverse and intensive workload of quantum-classical models, such as AQCNN. To optimize resource utilization, this work employs a systolic array-inspired parallel processing strategy, motivated by the model in [13], to distribute AQCNN tasks across multiple processing arrays. Utilizing a single array increases both energy consumption and computational cost. Therefore, AQCNN tasks are shared across multiple arrays to accelerate execution while reducing energy overhead. The TGPU architecture supports five distinct processing states during the AQCNN computation, as presented in Table I.

TABLE I. COMPARATIVE STUDY

State	Description
Initialization	Prepares the TGPU system and initializes qubits, typically to the $\|0\rangle$ state.
Execution (Active)	Applies quantum gates and classical operations to perform the AQCNN computation.
Idle (Non-Active)	Maintains qubits in a passive state (e.g., superposition or entangled), awaiting further tasks.
Measurement (Readout)	Measures qubit states, collapsing superpositions into classical outputs (0 or 1).
Off	Powers down the architecture; no computation occurs.

Using the off and initialization states frequently causes significant overhead as a result of the time required to power on/off and reconfigure arrays. To minimize this, the idle state is preferred during non-computational periods. Idle arrays maintain qubit readiness with minimal energy usage and can resume execution quickly, reducing the overall training time for AQCNN. By intelligently leveraging idle states and distributing computation across larger arrays (instead of numerous smaller ones), the proposed method ensures faster convergence and improved classification accuracy, as discussed in the next section.

III. RESULTS AND DISCUSSION

This section studies the performance of the AQCNN model over other recently presented Q-CNN approaches, such as HQC-CNN [1], SQ-DQNN [2], REQ-CNN [5], and QPTCNN [7]. The QSAM simulator was used to design a systolic array-parallel computation model [13]. In addition, the PennyLane library was used, employing TensorFlow and the PyTorch library, and for non-noisy simulations, the Qulacs qubit simulator was used as a plugin to PennyLane. The experiments were conducted utilizing subsets of MNIST data, which have multi- and binary classification tasks. The MNIST dataset [14] is used as a benchmark to evaluate various ML systems for image classification. As quantum approaches are in their early phases, the MNIST dataset provides a challenge because of fewer performance requirements with fewer parameters. Moreover, for further evaluation, the FMNIST [15] dataset was also considered. The FMNIST dataset also consists of multi- and binary classification tasks. As FMNIST data are sometimes thought to be more complicated than MNIST data, it presents a suitable challenge that the low-parameter system can successfully handle.

A. Classification Performance: Case Study 1

Figure 3 evaluates the performance of AQCNN compared to traditional CNN and Quantum-based CNN models. For comparison purposes, both traditional CNN and QCNN models were implemented alongside AQCNN. On MNIST, the CNN model achieved 94.28% accuracy, while QCNN slightly outperformed it with 94.61%. The proposed AQCNN model demonstrated superior performance, achieving 95.84% accuracy. Similarly, on FMNIST, the CNN and QCNN models achieved accuracies of 94.61% and 93.12%, respectively. Again, AQCNN outperformed both, achieving 95.73%. These results highlight the effectiveness and robustness of AQCNN across different datasets, demonstrating its potential for improved classification accuracy by leveraging the strengths of both classical and quantum computing paradigms.

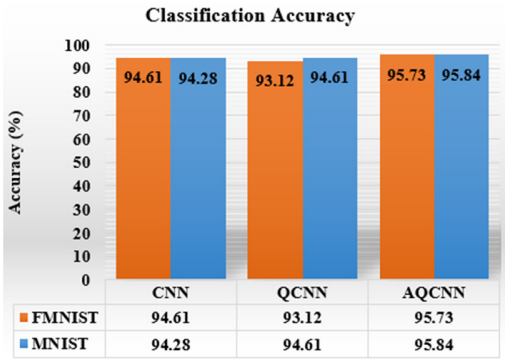

Fig. 3. Classification accuracy of AQCNN compared with traditional CNN and QCNN considering MNIST and FMNIST.

B. Classification Performance: Case Study 2

This section presents a comparative analysis of the performance of AQCNN with the SQ-DQNN approach [2], using both binary and multi-class classification tasks on the FMNIST and MNIST datasets, as presented in Figures 4 and 5, respectively. For consistency, the same datasets and class configurations were used for AQCNN. In the binary classification scenario, AQCNN achieved 95.68% accuracy on FMNIST, significantly outperforming SQ-DQNN, which attained 89.5%. Similarly, for the MNIST binary classification task, AQCNN achieved 95.88%, surpassing SQ-DQNN's 94.6%.

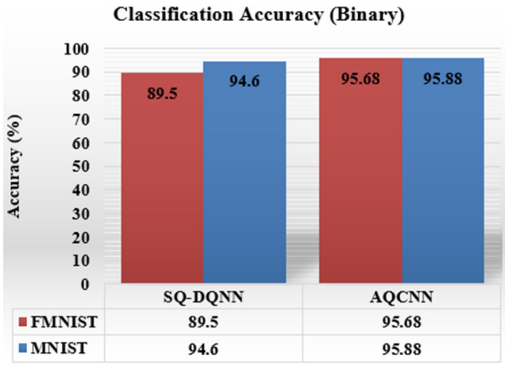

Fig. 4. Classification accuracy of AQCNN compared to the SQ-DQNN approach for binary classification using FMNIST and MNIST.

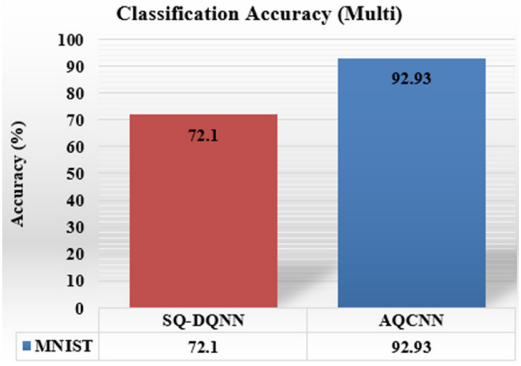

Fig. 5. Classification accuracy of AQCNN compared to the SQ-DQNN approach for multiclass classification using MNIST.

In the multi-class classification setting on the MNIST dataset, AQCNN again demonstrated superior performance with 92.93% accuracy, compared to just 72.1% achieved by SQ-DQNN. These results show the enhanced classification capabilities of AQCNN in both binary and multi-class tasks, affirming its advantage over existing QNN approaches.

C. Comparative Study

The comparative results in Table II highlight the superior performance of the AQCNN model in image classification tasks using the FMNIST and MNIST datasets. Compared to models from recent studies, AQCNN consistently demonstrates higher accuracy and robustness. In [1], various classical and quantum-inspired models were evaluated on the MNIST dataset, including CNN (52.2%), QNN (63.9%), QCNN (65%), ResNet (65.9%), DenseNet (66.3%), and QC-CNN (67.2%). These models significantly underperformed relative to the proposed AQCNN, which achieved an accuracy of 95.88% on the same dataset, highlighting a substantial improvement of over 28% compared to the best-performing model (QC-CNN) in [1]. Similarly, when compared to the REQ-CNN [5], which achieved accuracies of 92.94% on FMNIST and 92.84% on MNIST, AQCNN outperforms it with 95.68% on FMNIST and 95.88% on MNIST. This demonstrates not only a notable increase in accuracy but also the robustness and efficiency of AQCNN across both datasets. Furthermore, in [7], two versions of the QPTCNN were evaluated. Model A achieved 83.5% on FMNIST and 91% on MNIST, while Model B achieved 82% on FMNIST and 91% on MNIST. Although QPTCNN models showed decent performance, they still lag behind the AQCNN model by a significant margin, with differences of over 12% on FMNIST and nearly 5% on MNIST. These comparisons clearly establish that the AQCNN not only surpasses traditional and quantum-enhanced classical models but also outperforms recent QNN approaches. The improvements are attributed to the effective integration of amplitude encoding, parameterized quantum circuits, and hybrid processing strategies used in AQCNN. These design elements enable better feature extraction and classification performance while optimizing quantum resource usage, making AQCNN a highly promising solution for real-world quantum image classification applications.

TABLE II. COMPARATIVE STUDY

Ref	Year	Model	FMNIST Accuracy	MNIST Accuracy
[1]	2024	CNN	-	52.2
		QCNN	-	65
		QNN	-	63.9
		ResNet	-	65.9
		DenseNet	-	66.3
		QC-CNN	-	67.2
[5]	2024	REQ-CNN	92.94	92.84
[7]	2025	QPTCNN (Model A)	83.5	91
		QPTCNN (Model B)	82	91
	Proposed	AQCNN	95.68	95.88

IV. CONCLUSION

This study presented an AQCNN approach for image classification tasks. A key limitation in existing quantum-based CNN architectures is the restriction on the number of qubits, which is typically constrained by the volume and nature of the encoded input data. To address this challenge, the AQCNN model introduced a classical encoding layer before the quantum embedding layer, enabling more flexible qubit utilization and allowing the system to scale effectively. Furthermore, a classical fully connected layer was integrated after the quantum measurement stage to refine the output and improve classification performance. The AQCNN was evaluated using the MNIST and FMNIST datasets, demonstrating significant improvements over both classical and quantum benchmark models. Specifically, AQCNN achieved 95.88% accuracy on MNIST and 95.68% on FMNIST, outperforming models such as CNN, QCNN, QNN, ResNet, DenseNet, and several advanced quantum CNN architectures, including REQ-CNN and QPTCNN. These results show the effectiveness of AQCNN in achieving high classification accuracy while maintaining computational efficiency, due to its low-complexity design and hybrid quantum-classical structure. The AQCNN model presents a promising advancement in quantum-assisted image classification, offering a scalable and high-performing solution suited to the current NISQ era. Future research will focus on developing enhanced quantum kernels and a more sophisticated systolic array architecture to tackle more complex, high-dimensional remote sensing and real-world data with increased spatial and spectral resolution.

REFERENCES

[1] F. Fan, Y. Shi, T. Guggemos, and X. X. Zhu, "Hybrid Quantum-Classical Convolutional Neural Network Model for Image Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 12, pp. 18145–18159, Dec. 2024, https://doi.org/10.1109/TNNLS.2023.3312170.

[2] P. Easom-McCaldin, A. Bouridane, A. Belatreche, R. Jiang, and S. Al-Maadeed, "Efficient Quantum Image Classification Using Single Qubit Encoding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 1472–1486, Feb. 2024, https://doi.org/10.1109/TNNLS.2022.3179354.

[3] S. Moradi, C. Brandner, M. Coggins, R. Wille, W. Drexler, and L. Papp, "Error mitigation for quantum kernel based machine learning methods on IonQ and IBM quantum computers." arXiv, 2022, https://doi.org/10.48550/ARXIV.2206.01573.

[4] N. Ji, R. Bao, Z. Chen, Y. Yu, and H. Ma, "Hybrid Quantum Neural Network Image Anti-Noise Classification Model Combined with Error Mitigation," *Applied Sciences*, vol. 14, no. 4, Feb. 2024, Art. no. 1392, https://doi.org/10.3390/app14041392.

[5] Y. Song, J. Li, Y. Wu, S. Qin, Q. Wen, and F. Gao, "A resource-efficient quantum convolutional neural network," *Frontiers in Physics*, vol. 12, Apr. 2024, Art. no. 1362690, https://doi.org/10.3389/fphy.2024.1362690.

[6] Y. Li, R. G. Zhou, R. Xu, J. Luo, and W. Hu, "A quantum deep convolutional neural network for image recognition," *Quantum Science and Technology*, vol. 5, no. 4, Jul. 2020, Art. no. 044003, https://doi.org/10.1088/2058-9565/ab9f93.

[7] L. Hai, C. Liang, H. Yaming, Y. Wenli, and S. Fengquan, "An Improved Convolutional Neural Networks: Quantum Pseudo-Transposed Convolutional Neural Networks," *IEEE Access*, vol. 13, pp. 37108–37117, 2025, https://doi.org/10.1109/ACCESS.2025.3543164.

[8] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, Sep. 2014, https://doi.org/10.1038/nphys3029.

[9] T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification," *Quantum Machine Intelligence*, vol. 4, no. 1, Jun. 2022, Art. no. 3, https://doi.org/10.1007/s42484-021-00061-x.

[10] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, Sep. 2018, Art. no. 032309, https://doi.org/10.1103/PhysRevA.98.032309.

[11] F. Vatan and C. Williams, "Optimal quantum circuits for general two-qubit gates," *Physical Review A*, vol. 69, no. 3, Mar. 2004, Art. no. 032315, https://doi.org/10.1103/PhysRevA.69.032315.

[12] Y. Liang, W. Peng, Z. J. Zheng, O. Silvén, and G. Zhao, "A hybrid quantum–classical neural network with deep residual learning," *Neural Networks*, vol. 143, pp. 133–147, Nov. 2021, https://doi.org/10.1016/j.neunet.2021.05.028.

[13] T. Adiono, G. Meliolla, E. Setiawan, and S. Harimurti, "Design of Neural Network Architecture using Systolic Array Implemented in Verilog Code," in *2018 International Symposium on Electronics and Smart Devices (ISESD)*, Bandung, Indonesia, Oct. 2018, pp. 1–4, https://doi.org/10.1109/ISESD.2018.8605478.

[14] Li Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov. 2012, https://doi.org/10.1109/MSP.2012.2211477.

[15] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms." arXiv, 2017, https://doi.org/10.48550/ARXIV.1708.07747.