[1]Bhaskar Marapelli
[2]Hari Prasad Gandikota
[3]K S Ranadheer Kumar
[4]Sruthi Nath C
[5]Ch. Anil Carie
[6]Gandla Shivakanth

Quantum Convolutional Neural Networks for Line Orientation Classification in Pixelated Images

Abstract: - Quantum Convolutional Neural Networks (QCNNs) offer a promising avenue for image classification tasks due to their potential to leverage quantum properties for enhanced computational capabilities. In this paper, we explore the application of QCNNs for line orientation classification in pixelated images. Specifically, we investigate the differentiation between horizontal and vertical lines, a fundamental task in image processing and computer vision. We propose a QCNN architecture tailored to this task, leveraging quantum convolutional layers to extract features from pixelated images and classify line orientations. We demonstrate the effectiveness of our approach through experimental evaluation on benchmark datasets, comparing the performance of QCNNs with different optimizers. Our study integrated the QCNN operator and optimizer into Qiskit Machine Learning's Neural Network Classifier, leveraging quantum computing techniques for classification tasks. Our QCNN model demonstrated a training accuracy of 71.43% and a test accuracy of 60.0%. Noteworthy observations include the failure of the SPSA optimizer to converge within the designated iterations, requiring twice the iterations compared to the COBYLA optimizer for convergence.

Keywords: Quantum Convolutional Neural Networks (QCNNs), Image classification, Line orientation classification, Image processing, Computer vision, Qiskit Machine Learning, Neural Network Classifier.

I. INTRODUCTION

In recent years, quantum computing has attracted consider able attention for its potential to revolutionize computational capabilities, offering the promise of solving complex problems at an unprecedented speed, far surpassing the capabilities of classical computers [1]. This heightened interest has led to the emergence of Quantum Machine Learning (QML), a dynamic field that combines the foundational principles of quantum computing with traditional machine learning techniques to tackle real-world challenges with enhanced efficiency and effectiveness [2], [3]. Within the realm of QML, researchers have developed a variety of novel algorithms tailored to harness the unique properties of quantum systems. Examples include the quantum kernel estimator, which leverages quantum computations to enhance kernel-based machine learning methods [4], and Variational quantum circuits (VQCs) are employed to optimize ma chine learning models through variational techniques.

[1] *Bhaskar Marapelli: Associate Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India, bhaskarmarapelli@gmail.com.

[2] Hari Prasad Gandikota, Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad- 500075, Telangana, India, prasadghari4@gmail.com.

[3] K S Ranadheer Kumar: Assistant Professor, Department of Computer Science and Engineering(Data Science), CVR college of engineering, Hyderabad, India, ranadheer.k.s@gmail.com.

[4] Sruthi Nath C: Assistant Professor, Velammal Engineering College, Autonomous Institution - Affiliated to Anna University, sruthinath@velammal.edu.in.

[5] Ch. Anil Carie: Assistant Professor, Department of Computer Science and Engineering, SRM University, Amaravathi, AP, India, carieanil@gmail.com.

[6] Gandla Shivakanth: Associate Professor, Koneru Lakshmaiah Education Foundation, Hyderabad- 500075, Telangana, India, shvkanth0@gmail.com.

Copyright © JES 2024 on-line: journal.esrgroups.org

These circuits utilize quantum amplitudes to encode data and have been applied to various applications such as combinatorial optimization, quantum chemistry simulation, and quantum machine learning [5].

By using the special abilities of quantum computers, QML can help us process largescale datasets much faster and tackle computationally demanding tasks more accurately [6]. Models like QCNNs and Variational Quantum Algorithms (VQAs) have shown they can do things classical computers struggle with. They use quantum circuits and quantum data to make smarter algorithms and improve how machines learn [7]. Convolutional neural networks (CNNs) have emerged as highly effective architectures for classification tasks, particularly in image recognition [8]. A CNN typically comprises a series of interleaved layers, each processing images to generate intermediate two-dimensional arrays of pixels known as feature maps. These feature maps are computed through convolution layers, where new pixel values are derived from linear combinations of neighboring pixels in the preceding map. Pooling layers subsequently reduce feature map sizes, often by selecting maximum values from contiguous pixels, followed by activation functions [9]. As the size of the feature map diminishes through successive convolution and pooling layers, the ultimate output is determined by a fully connected layer. This layer's weights and function are optimized through training on extensive datasets, enabling the network to learn discriminative features effectively. Notably, the number of convolution and pooling layers, along with the size of weight matrices, remains fixed for a given Convolutional Neural Network (CNN) architecture, serving as hyperparameters. A distinctive characteristic of CNNs lies in their translationally invariant convolution and pooling layers, which maintain a constant number of parameters across different regions of the input data. This property facilitates sequential reduction in data size and imparts a hierarchical structure to the network, enabling it to extract increasingly abstract features as information flows through successive layers[10].

QCNNs function in a manner akin to classical CNNs. The initial step involves encoding a pixelated image into a quantum circuit, a process facilitated by selecting an appropriate feature map. For instance, popular choices like Qiskit's ZFeatureMap or ZZFeatureMap are commonly utilized for this purpose. This encoding procedure lays the groundwork for subsequent quantum operations, enabling the extraction of features from the image data in a quantum-compliant format. Subsequently, alternating convolutional and pooling layers are applied to reduce the dimensionality of the quantum circuit until only one qubit remains, enabling image classification through qubit measurement. Quantum Convolutional Layers consist of two qubit unitary operators to identify relationships between qubits. Quantum Pooling Layers, unlike classical counterparts, reduce qubit numbers by selectively performing operations until a specific threshold, after which certain qubits are disregarded, defining the pooling layer. Each QCNN layer incorporates parametrized circuits, allowing adjustment of output results by tuning layer parameters during training to minimize the loss function [11].

In this study, we undertake an investigation into both classical and quantum CNNs. While traditional CNNs have demonstrated efficacy across a spectrum of classification tasks, our endeavor introduces a novel paradigm by integrating quantum circuits, quantum convolutional, and pooling layers to formulate a QCNN. This innovative QCNN architecture represents a departure from conventional approaches, allowing us to harness the unique principles of quantum mechanics for image classification endeavors. By leveraging quantum circuits and specialized quantum layers, our objective is to augment the capabilities of traditional CNNs and elucidate the potential advantages conferred by quantum computing within the domain of image recognition.

II. RELATED WORK

The introduction of quantum computing marks a significantadvancement in computational theory, offering the potentialto solve complex problems at an exponential speed compared to classical computers. QML emerges as a promising field, combining the principles of quantum computing with traditional machine learning techniques to address realworld challenges more efficiently. The works by Tychola et al. (2023) and Winker et al. (2023) [12], [13] explore QML,a field that merges quantum computing with conventionalmachine learning methods to tackle realworld challenges withincreased efficiency.

Current research in quantum machine learning encompassesa diverse range of algorithms and methodologies aimed atleveraging the unique properties of quantum systems to enhance computational tasks. Various approaches, such as quantum kernel methods and variational quantum circuits, have demonstrated promising results across different applications, including pattern recognition and classification tasks.The workby Raubitzek et al. (2023)

[14] examines the applicabilityof QML algorithms, such as Quantum Support Vector Machines (QSVM). These algorithms have demonstrated superiorperformance compared to classical counterparts, particularly on complex datasets. The paper by Thomas et al. (2023) focuses on the analysis, benefits, limitations, and potentialapplications of quantum neural networks, and provides insightsinto the broader landscape of quantum machine learning research [15]. In the research article by Senokosov et al. [16], they introduced two hybrid quantumclassical modelsfor image classification. By dividing the quantum part into multiple parallel variational quantum circuits, they achieved efficient neural network learning. Remarkably, this approach achieved an accuracy of over 99% on the MNIST dataset. Theresearch paper by Zhou et al. [17] focuses on proposing a newquantum neural network model for quantum neural computingusing single qubit operations and measurements on real-world quantum systems.

In parallel, classical CNNs have established themselves as apowerful architecture for image recognition and classification.These networks consist of multiple layers of image processing,including convolutional and pooling layers, followed by fully connected layers for classification. The works by Celeghin et al. (2023), Ramprasath et al. (2018), and Huang et al. (2022) [8], [18], [19] discuss the architecture of CNNs for image classification. Building upon this foundation, recent works have focused on developing QCNNs by integrating quantum circuits and specialized quantum layers into the architecture. QCNNs aim to harness the quantum advantage to improve image classification tasks by exploiting quantum principles such as superposition and entanglement. Motivated by the architectureof classical CNNs, Cong et al. (2019) [11] introduce a QCNN circuit model. This QCNN model extends the key propertiesof classical CNNs to the quantum domain, leveraging quantumprinciples to enhance image processing and classification tasks. The paper by Meedinti et al. (2023) [20] focuses on the evaluation of QCNNs in comparison to classical CNNs and Artificial/Classical Neural Network (ANN) models for object detection and classification tasks.

By combining classical and quantum approaches, researchers seek to push the boundaries of machine learning and explore the potential of quantum computing in solving computationally intensive tasks more effectively. These effortsrepresent an exciting frontier in the intersection of quantum computing and machine learning, with the potential to unlock new possibilities in image recognition and beyond.

III. ARCHITECTURE OF A CLASSICAL CONVOLUTIONAL NEURAL NETWORK (CNN)

A Convolutional Neural Network (CNN), often referred to as ConvNet, emerges as a specialized variant of deep learning algorithms meticulously crafted to tackle tasks necessitating object recognition with precision. This deep learning architecture, characterized by convolutional layers, a fully connected layer, a pooling layer, and associated weights, epitomizes the cornerstone of modern image processing and recognition systems. Notably, CNNs boast a leaner parameter count compared to their deep, feed-forward neural network counterparts, rendering them particularly appealing for intricate deep learning tasks [21]. Leveraging their robust architecture, CNNs find extensive applications spanning diverse domains such as image classification, segmentation, object detection, video processing, natural language processing, and speech recognition [22]. The architectural layout of a CNN, delineated in Figure 1, illustrates the intricate interplay of its constituent layers, underscoring the network's prowess in discerning intricate patterns and features within complex datasets.

a) Input Layer

The input layer comprises the raw pixel values of the image. Let X represent the input image, where X is a matrix of size N×M×C, with N denoting the height, M representing the width, and C indicating the number of channels (such as RGB channels for a color image).

b) Convolutional Layer

The convolutional layer applies a set of learnable filters, also known as kernels, to the input image in order to extract features. Let F represent the set of filters, each with dimensions k times k times C, where k is the kernel size and C is the number of channels. The convolution operation between the input image X and the filter F is computed as follows:

$$Z(i,j) = \sum_{m=1}^{k}\sum_{n=1}^{k}\sum_{c=1}^{C} X(i+m-1, j+n-1, C)\ x\ F\ (m,n,c)$$

Here, Z represents the feature map, and i and j iterate over the spatial dimensions of the feature map.

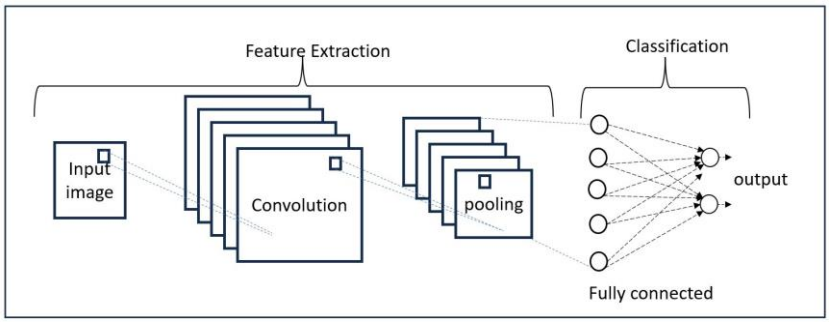

Fig. 1. Architecture of a Classical Convolutional Neural Network (CNN)

c) Activation Function

After convolution, an activation function is applied elementwise to the feature map to introduce nonlinearity. Common activation functions include ReLU (Rectified Linear Unit),sigmoid, and tanh.

d) Pooling Layer

The pooling layer reduces the spatial dimensions of the feature map while retaining important information. Max pooling is a common pooling technique, where the maximum valuewithin a small window is selected. Let *P* represent the pooling operation, and *S* represent the stride (the amount by which the window shifts). Mathematically, max pooling is computed as follows:

$$P(i,j) = \max_{m,n} Z(i\ x\ S\ +\ m, j\ x\ S\ +n)$$

e) Fully Connected Layer

Following the application of multiple convolutional and pooling layers within the Quantum Convolutional Neural Network (QCNN), the resultant feature maps undergo a crucial transformation. These feature maps, which capture important spatial features extracted from the input data, are flattened into a one-dimensional vector. This flattening process condenses the spatial information contained within the feature maps into a format suitable for further processing, this flattened vector is fed into fully connected layers, which play a pivotal role in the classification task. Within these fully connected layers, the input vector interacts with weight parameters represented by the weight matrix W and a bias term denoted by the bias vector b. The output of the fully connected layer is computed through a series of matrix operations, where the input vector is multiplied by the weight matrix and added to the bias vector. This computation yields a transformed representation of the input data, facilitating the extraction of higher-level features and enabling more intricate classification tasks.

$$Y = Activation(W.X_{flat} + b)$$

Here, X_{flat} represents the flattened feature vector.

IV. QUANTUM CONVOLUTIONAL NEURAL NETWORKS (QCNN)

When compared to traditional convolutional neural networks (CNNs), the behavior of quantum convolutional neural networks (QCNNs) is similar. Usually, the procedure starts with encoding a pixelated image into a quantum circuit using a chosen feature map, like ZFeatureMap or ZZFeatureMap from Qiskit, or other choices from the circuit library. This encoding technique is essential because it converts the picture data into a quantum representation that can be used with quantum computing. As described in the following section, after the image

is encoded into the quantum circuit, a sequence of alternating convolutional and pooling layers is utilized. These layers are essential to the feature extraction procedure, which finds and extracts features from the quantum representation that are pertinent to the classification problem. The dimensionality of the quantum circuit is gradually decreased by applying these alternating layers, which eventually results in the representation of the image in terms of a single qubit.

After the input image is reduced to this level of dimensionality, the input image must be classified. By measuring the output state of the lone surviving qubit, this is accomplished. The measurement's result offers useful data for identifying the class or category that the input image falls into. To put it simply, the procedure involves converting a pixelated image into a classifier based on the quantum state of a single qubit after a series of quantum operations.

a) Quantum Convolutional Layer:

The Quantum Convolutional Layer comprises a sequence of two-qubit unitary operators denoted by U, which discern and establish relationships among the qubits within our circuit. These unitary gates are described by the equation:

$$|\psi'\rangle = U|\psi\rangle$$

Here,

$|\psi\rangle$ represents the quantum state of the circuit before applying the unitary operator, and

$|\psi'\rangle$ is the state after the operator's application.

b) Quantum Pooling Layer:

In the Quantum Pooling Layer, we aim to reduce the number of qubits by performing operations on each qubit until reaching a specific threshold, after which we selectively discard certain qubits within a designated layer. The layers where we cease operations on specific qubits are termed as our 'pooling layer'. Further elaboration on the pooling layer is provided in the subsequent section. Within the Quantum Convolutional Neural Network (QCNN), each layer consists of parametrized circuits, allowing us to modify the output result by adjusting the parameters associated with each layer. During the training phase of our QCNN, these parameters undergo adjustment to minimize the loss function associated with the network.

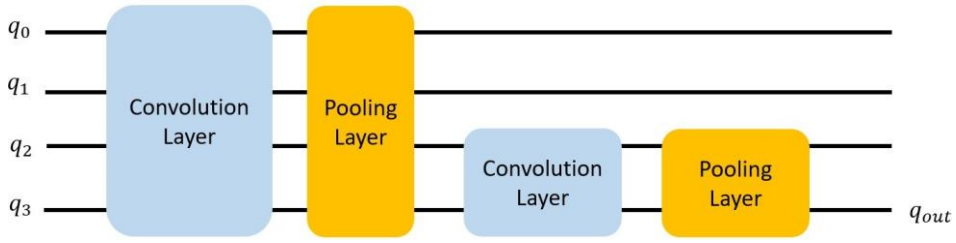

Fig. 2. Four qubit QCNN

V. EXAMPLE QUANTUM CONVOLUTIONAL NEURAL NETWORK (QCNN)

Let us examine a hypothetical example of a four-qubit QCNN, as shown in Figure 2. All four qubits are subjected to operation by the first Convolutional Layer, which produces a quantum state called $|\psi_1\rangle$.

a) First Pooling Layer:

The first pooling layer serves to decrease the dimensionality of the QCNN from four qubits to two qubits by disregarding the first two. Following the application of the first pooling layer, the resulting quantum state is denoted as $|\psi_2\rangle$.

b) Second Convolutional Layer:

Subsequently, the second Convolutional layer is employed to identify features between the two qubits that remain in use within the QCNN. The quantum state obtained after applying the second Convolutional Layer is represented as $|\psi_3\rangle$.

c) Second Pooling Layer:

A subsequent pooling layer is applied, further reducing the dimensionality from two qubits to one, thereby yielding our output qubit. The quantum state resulting from the application of the second pooling layer is denoted as $|\psi_4\rangle$.

The final output qubit, represented as $|\psi_{output}\rangle$, is derived from $|\psi_4\rangle$ through measurement.

Fig. 3. Sample images from synthetic dataset.

V. EXPERIMENTAL SETUP

A classical Convolutional Neural Network (CCNN) typically includes both convolutional and pooling layers. The definition of these QCNN convolutional and pooling layers is given in terms of gates that are applied to a quantum circuit that uses qubits. The parameters in each layer of the QCNN will be changed during training in order to minimize the loss function. This training process aims to refine the QCNN's ability to classify between horizontal and vertical lines. To define the Convolutional Layers, a parametrized unitary gate was determined first, which was used to create the convolutional and pooling layers. Following the definition of these unitaries, a function was formulated to implement the convolutional layer within the QCNN. The pooling layer in QCNN pairs qubits, applies a two qubitunitary operation to each pair, and then disregards one qubit from each pair for subsequent layers. This process consolidates information between qubits, enhances feature representation, and reduces circuit dimensionality. Once applied, no further operations or measurements are performed on the discarded qubits.

The pooling layer orchestrates a transformative process whereby the dimensions of our quantum circuit, originally composed of N qubits, undergo a reduction to N/2. This reduction in dimensionality is a pivotal step within the QCNN, enabling more efficient processing and extraction of key features from the input data.

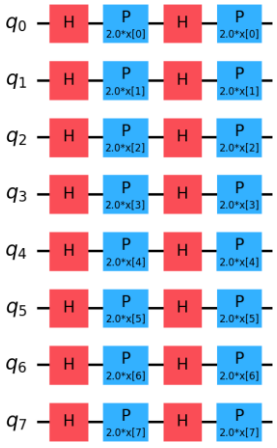

Fig. 4. Z feature Map

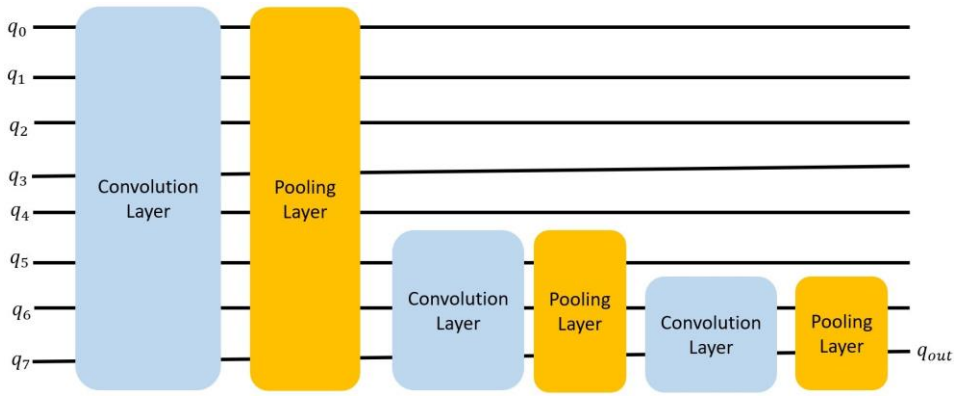

Fig. 5. QCNN from eight qubits to one

a) Dataset

A synthetic dataset is created for training and testing the QCNN, where each sample consists of a 2x4 pixelated image containing either a horizontal or vertical line alongside a noisybackground. In Figure 3, a collection of sample images is presented, each meticulously crafted to depict either a vertical or horizontal line. These images serve as illustrative examples showcasing the fundamental elements crucial for line orientation classification tasks within image processing and computer vision. Through these samples, the distinct characteristics and orientations of lines are visually represented, laying the groundwork for the subsequent analysis and exploration of line orientation classification methodologies.

b) Model building.

The Model building steps involves constructing the Quantum QCNN by incorporating alternating pooling and convolutional layers. Since the dataset comprises images with 8 pixels,QCNN will consist of 8 qubits. To encode the dataset intothe QCNN, a feature map(Z feature) is applied, see Figure 4.The QCNN function is defined as incorporating three setsof alternating convolutional and pooling layers. The layers incorporated in the architecture of the QCNN are meticulously crafted with the specific aim of reducing the dimensionality of the network from eight qubits down to a single qubit, as visually depicted in Figures 5 and 6. This reduction in dimensionality is a crucial aspect of the network's design, facilitating streamlined processing and analysis of input data. In the context of classifying the image dataset containing horizontal and vertical lines, a pivotal step involves measuring the expectation value of the Pauli Z operator associated with the final qubit. This measurement serves as a decisive criterion for discerning the orientation of lines within the images, thus enabling accurate classification based on the quantum state of the network's output.

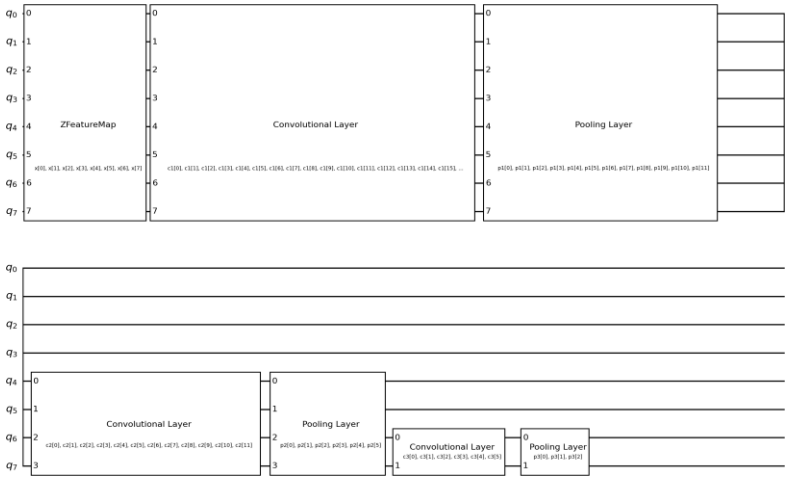

Fig. 6. QCNN Model Prepared

Following the measurement of the expectation value of the Pauli Z operator associated with the final qubit, the inference process hinges on the interpretation of the obtained value, which can be either +1 or -1. This interpretation serves as the decisive factor in determining whether the input image contained a horizontal or vertical line. The QCNN leverages this inference mechanism to make accurate classifications based on the quantum state of its output. Moreover, it's noteworthy that the construction of the QCNN is facilitated using Qiskit, a Python library dedicated to quantum computing. This integration underscores the synergy between classical and quantum computing paradigms, harnessing the power of both to realize advanced machine learning models capable of tackling complex classification tasks with enhanced efficiency and accuracy. It begins with initializing a feature mapusing the ZFeatureMap method, which encodes classical datainto a quantum state suitable for processing by the quantum neural network. The ansatz, representing the parameterizedpart of the quantum circuit, is then initialized with 8 qubits.Convolutional layers are subsequently added to the ansatzcircuit using the conv layer function, which applies quantumgates to recognize and extract features from input quantumstates. Additionally, pooling layers are integrated into theansatz circuit to reduce the dimensionality of quantum statesby combining information from neighboring qubits. Finally,the feature map circuit and the ansatz circuit are merged into asingle quantum circuit, representing the complete architectureof the quantum convolutional neural network, Figure 6 describes the QCNN complete architecture using Qiskit library.

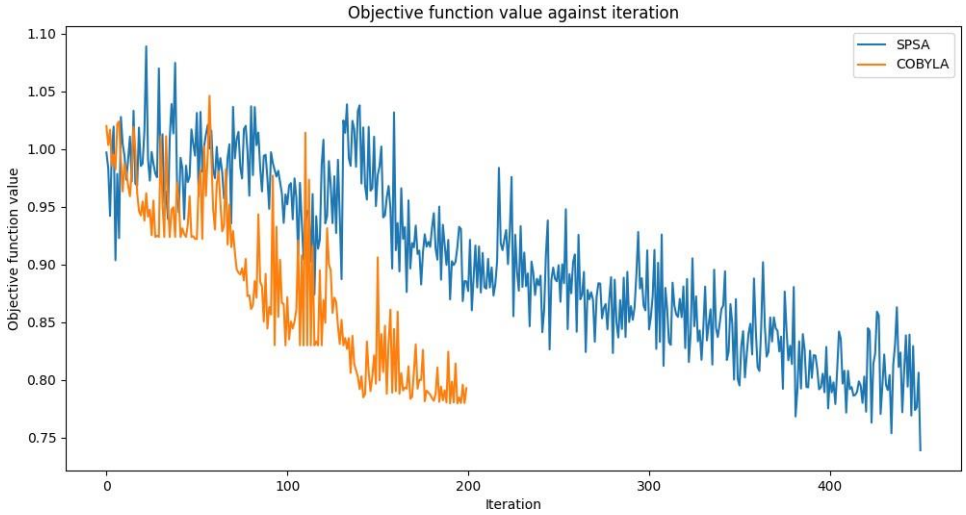

Fig. 7. Objective function value vs iteration

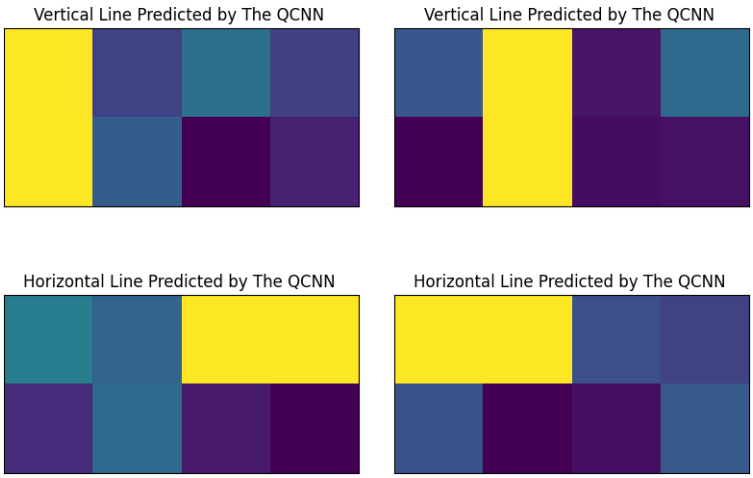

Fig. 8. Predictions Made by QCNN Model

VI. RESULTS AND DISCUSSION

The QCNN operator and optimizer are incorporated into Qiskit Machine Learning's Neural Network Classifier. Themodel is trained using the training dataset along with corresponding labels.

During the training phase of the model, a pivotal aspect involves leveraging the expectation value of the Pauli Z qubit associated with the final qubit as a measurement metric. This expectation value serves as a crucial indicator, where a resultant value of +1 or -1 corresponds respectively to the presence of a vertical or horizontal line within the input image. This utilization of quantum measurement outcomes not only enables QCNN to discern line orientations effectively but also facilitates the learning process by providing valuable feedback for model optimization. To ensure effective monitoring and analysis of the training progress, a callback function is employed. This callback function plays a vital role in dynamically tracking and plotting the loss function throughout the training iterations. By visualizing the loss function trends over time, researchers and practitioners gain insights into the model's learning dynamics, enabling informed adjustments and optimizations to enhance training efficacy and convergence. This iterative monitoring process is essential for iteratively refining the QCNN architecture and parameters to achieve optimal performance in line orientation classification tasks. The results are analyzed by applying COBYLA (Constrained Optimization BY Linear Approximations) and SPSA (Simultaneous Perturbation Stochastic Approximation) optimizers.

a) COBYLA:

COBYLA is an optimization algorithm that iteratively adjusts the parameters to minimize an objective function subject to constraints. The algorithm approximates the objective function and constraints using linear approximations and searches for the optimal solution within the feasible region.

b) SPSA:

SPSA is a stochastic approximation method used for optimizing objective functions in the presence of noise or unknowngradients. The algorithm estimates the gradient of the objective function by perturbing the parameters stochastically and evaluating the objective function at the perturbed points.

In Figure 7, a plot depicting the relationship between the Objective function value and the iteration count is showcased. This visualization provides insights into the convergence behavior of the QCNN model during training. Additionally, the QCNN model achieved a training accuracy of 71.43% and a test accuracy of 60.0%. Notably, it was observed that the Simultaneous Perturbation Stochastic Approximation (SPSA) optimizer failed to converge within the designated number of iterations. Furthermore, the SPSA optimizer necessitated double the number of iterations compared to the COBYLA optimizer to achieve convergence. The plot in Figure 7 effectively illustrates the convergence behavior of the model with each optimizer function, offering valuable insights into the convergence time required by the model.

Figure 8 presents the predictions made by the model regarding the presence of horizontal and vertical lines within the images. This visualization offers a qualitative assessment of the model's performance in classifying line orientations within the image dataset.

VII. CONCLUSION

In conclusion, our study integrated the QCNN operator and optimizer into Qiskit Machine Learning's Neural Net work Classifier, leveraging quantum computing techniquesfor classification tasks. Through training the model on the provided dataset alongside corresponding labels, we employedthe expectation value of the Pauli Z qubit as a measurement, wherein +1 and -1 denoted vertical and horizontal line classifications, respectively. Our QCNN model achieved a training accuracy of 71.43% and a test accuracy of 60.0%. Notably, weobserved that the SPSA optimizer failed to converge within the designated iterations, requiring twice the iterations compared to the COBYLA optimizer for convergence. These findings underscore the importance of selecting an appropriate optimizer for quantum neural network training, as it significantly impacts convergence time and ultimately, model performance.We demonstrated the effectiveness of our approach through experimental evaluation on benchmark datasets, comparing theperformance of QCNNs on two optimizers.

ACKNOWLEDGEMENTS

We utilized the Qiskit framework, developed and maintained by IBM Corporation, in our research. Specifically, a section of our code was adapted from a Qiskit project © Copyright IBM Corporation 2017, 2024. We express our gratitude to the Qiskit development team for their contributions to quantum computing software development.

REFERENCES

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp.195–202, 2017.

[2] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," *Physical Review A*, vol. 94, no. 2,p. 022342, 2016.

[3] S. Srinivasan, C. Downey, and B. Boots, "Learning and inference in hilbert space with quantum graphical models," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[4] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, no. 1, p. 103, 2019.

[5] C.-C. J. Wang and R. S. Bennink, "Variational quantum regression al- gorithm with encoded data structure," *arXiv preprint arXiv:2307.03334*,2023.

[6] K. Chinzei, Q. H. Tran, K. Maruyama, H. Oshima, and S. Sato, "Splittingand parallelizing of quantum convolutional neural networks for learningtranslationally symmetric data," *arXiv preprint arXiv:2306.07331*, 2023.

[7] J. Lindsay and R. Zand, "A novel stochastic lstm model inspired by

quantum machine learning," in *2023 24th International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2023, pp. 1–8.

[8] A. Celeghin, A. Borriero, D. Orsenigo, M. Diano, C. A. M. Guerrero,

A. Perotti, G. Petri, and M. Tamietto, "Convolutional neural networks for vision neuroscience: significance, developments, and outstandingissues," *Frontiers in Computational Neuroscience*, vol. 17, 2023.

[9] P. Shruti and R. Rekha, "A review of convolutional neural networks, its variants and applications," in *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS)*. IEEE, 2023, pp. 31–36.

[10] P. Kocsis, P. Súkeník, G. Brasó, M. Nießner, L. Leal-Taixé, and I. Elezi, "The unreasonable effectiveness of fully-connected layers for low-data regimes," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1896–1908, 2022.

[11] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[12] K. A. Tychola, T. Kalampokas, and G. A. Papakostas, "Quantum machine learning—an overview," *Electronics*, vol. 12, no. 11, p. 2379, 2023.

[13] T. Winker, S. Groppe, V. Uotila, Z. Yan, J. Lu, M. Franz, and

W. Mauerer, "Quantum machine learning: Foundation, new techniques, and opportunities for database research," in *Companion of the 2023 International Conference on Management of Data*, 2023, pp. 45–52.

[14] S. Raubitzek and K. Mallinger, "On the applicability of quantum machine learning," 2023.

[15] R. K. Thomas, "Quantum neural networks: Bridging quantum computing and machine learning," *Asian Journal of Research in Computer Science*,vol. 16, no. 3, pp. 1–10, 2023.

[16] A. Senokosov, A. Sedykh, A. Sagingalieva, and A. Melnikov, "Quantum machine learning for image classification," *arXiv preprint arXiv:2304.09224*, 2023.

[17] M.-G. Zhou, Z.-P. Liu, H.-L. Yin, C.-L. Li, T.-K. Xu, and Z.-B. Chen, "Quantum neural network for quantum neural computing," *Research*, vol. 6, p. 0134, 2023.

[18] M. Ramprasath, M. V. Anand, and S. Hariharan, "Image classification using convolutional neural networks," *International Journal of Pure andApplied Mathematics*, vol. 119, no. 17, pp. 1307–1319, 2018.

[19] K. Huang, "Image classification using the method of convolutional neural networks," in *2022 IEEE Conference on Telecommunications, Optics and Computer Science (TOCS)*. IEEE, 2022, pp. 827–832.

[20] G. N. Meedinti, K. S. Srirekha, and R. Delhibabu, "A quantum convolu-tional neural network approach for object detection and classification," *arXiv preprint arXiv:2307.08204*, 2023.

[21] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutionalneural networks: analysis, applications, and prospects," *IEEE transac- tions on neural networks and learning systems*, 2021.

[22] P. Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K.Frisky, and Q. M. ul Haq, "Understanding of convolutional neural network (cnn): A review," *International Journal of Robotics and ControlSystems*, vol. 2, no. 4, pp. 739–748, 2022.