Classification of Oil Palm Trees Using Quantum Convolutional Neural Network (QCNN)

Desi Amirullah[1, a)], Lipantri Mashur Gultom[2]

[1,2]*Informatic Engineering, Politeknik Negeri Bengkalis, Riau, Indonesia*

[a)]Corresponding author: lipantri@polbeng.ac.id

Abstract. The oil palm trees (Elaeis guineensis) is an important commodity in the plantation industry, and the classification of its varieties is crucial for enhancing productivity and harvesting efficiency. This study aims to apply Quantum Convolutional Neural Network (QCNN) as a method for classifying oil palm trees. QCNN integrates quantum computing principles into the architecture of convolutional neural networks, allowing for more efficient and accurate data processing. The data used in this research includes oil palm plantations located in the coastal areas of Bengkalis Island. Data acquisition was performed using a DJI Phantom 4 Pro drone, capturing vertical images from above. The classification process utilized key features extracted from the images using the QCNN algorithm. The results of the experiments show that QCNN achieved a training accuracy of 94.7% and a testing accuracy of 92.5%. Thus, this research makes a significant contribution to the development of oil palm tree classification technology and opens new opportunities for the application of quantum algorithms in agriculture. These findings are expected to assist farmers and researchers in identifying and managing oil palm tree varieties more effectively, thereby supporting the sustainability and productivity of the oil palm industry as a whole.

Keywords: quantum, convolutional, neural network, oil palm trees

INTRODUCTION

Oil palm (Elaeis guineensis) is a strategic commodity in Indonesia's plantation industry, with Indonesia having the largest oil palm plantation area in the world, particularly in Bengkalis Regency. According to the 2023 data from the Central Bureau of Statistics (BPS) of Bengkalis Regency, the region boasts an oil palm plantation area of 399,783.81 hectares, with a production volume of 748,615.87 tons [1]. One of the main challenges in improving the productivity of oil palm plantations is the ability to accurately classify oil palm tree varieties. Proper classification is crucial for plantation management, from planting to harvesting, which ultimately contributes to increased yield and production efficiency. Coastal oil palm plantations in Bengkalis Regency face additional challenges, such as vulnerability to climate changes, including saltwater intrusion and temperature fluctuations, which affect the growth of oil palm trees. Moreover, the distinct soil characteristics and environmental conditions in coastal areas can influence the visual appearance of oil palm trees. Therefore, data collected from plantations in these regions must be carefully and accurately managed to address these variations. This necessitates the use of advanced technology capable of efficiently analyzing and classifying oil palm trees, especially for large areas like those in Bengkalis.

In recent years, technological advancements, particularly in the field of Artificial Intelligence (AI), have opened new opportunities for addressing this classification issue. One of the most promising recent methods in image processing and classification is the use of Quantum Convolutional Neural Networks (QCNN). QCNN combines quantum computing principles with the architecture of convolutional neural networks (CNN), which are well-known for their effectiveness in pattern recognition from image data [2]. To support the collection of plantation image data, this research utilizes the DJI Phantom 4 Pro drone to capture vertical images from above the plantation areas on Bengkalis Island. This drone enables efficient and accurate data collection, which is then processed using the QCNN algorithm to classify oil palm trees. However, processing large-scale image data requires a computational approach that is both efficient and accurate, making the application of Quantum Convolutional Neural Networks (QCNN) critical in this context [3][4][5].

Based on the background outlined above, the research questions for this study are as follows: How can Quantum Convolutional Neural Networks (QCNN) be applied to accurately classify oil palm trees in coastal plantation areas of Bengkalis Regency ? How effective are aerial images taken by the DJI Phantom 4 Pro drone in supporting the classification process using QCNN ? and How are the accuracy and efficiency results with several optimization methods in data classification training ? [6][7][8][9]

METHODS

In this research, the dataset used consists of two types, namely the palm tree dataset in the form of a collection of digital images taken from the DJI Phantom 4 Pro drone from several oil palm plantation locations in the coastal area of Bengkalis district along with the data samples:

FIGURE 1. oil palm tree samples

the second is a dataset of non-oil palm digital images and sample data:

FIGURE 2. samples are not oil palm trees

Each of these two datasets was then divided into two, namely a training dataset of 90 data and a testing dataset of 60 data.

Next, the design of the artificial neural network architecture using the quantum circuit approach consists of two main parts, namely the first part, namely the encoding part, which consists of 3 types as follows:

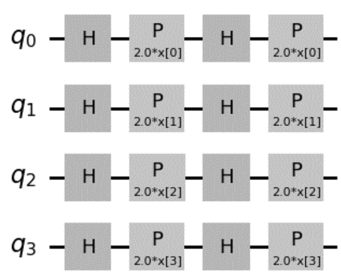

FIGURE 3. Encoding ZFeatureMap

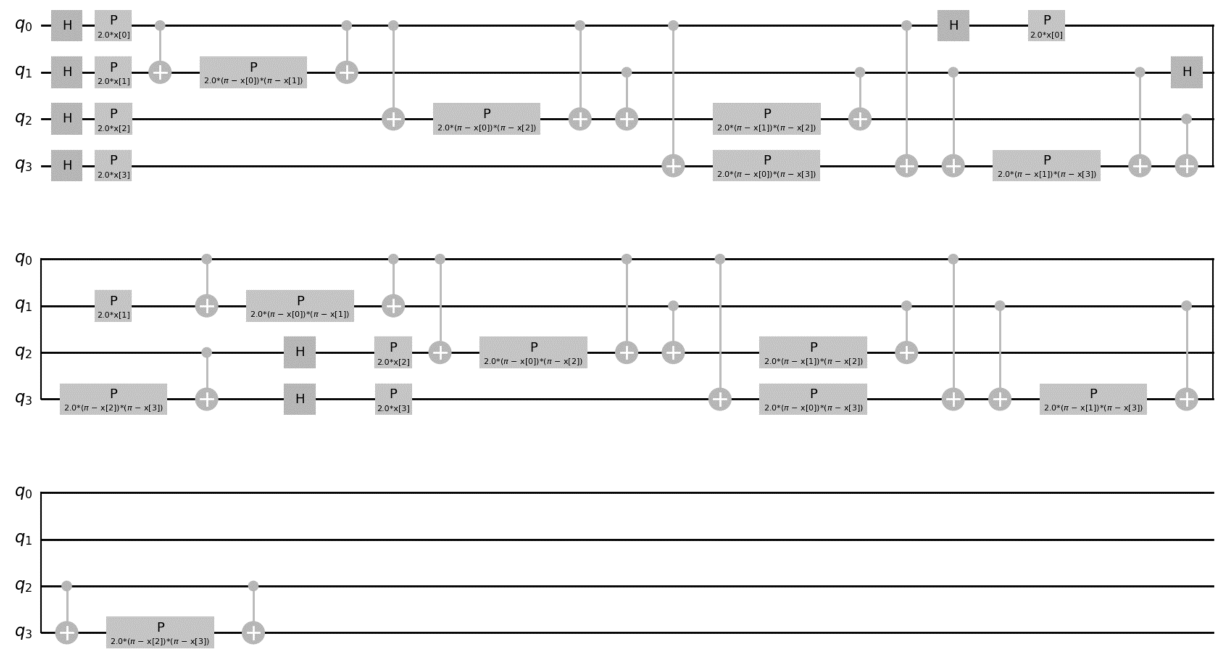

FIGURE 4. Encoding ZZFeatureMap

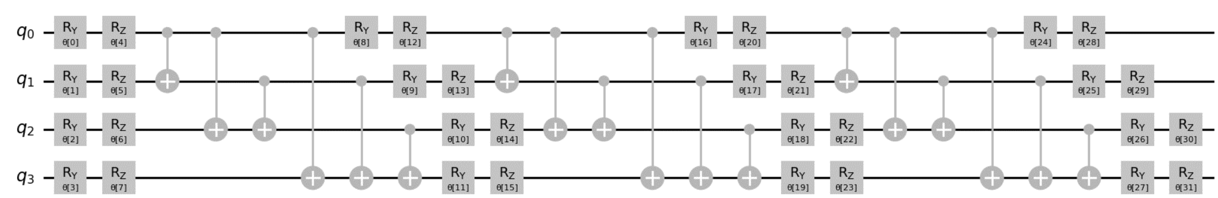

FIGURE 5. Encoding EfficientSU2

then the second part is the QCNN part as follows:

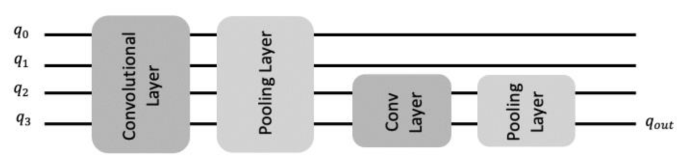

FIGURE 6. QCNN architecture [10]

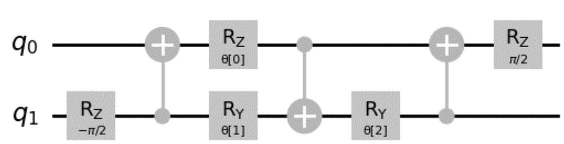

FIGURE 7. Convolutional Layer

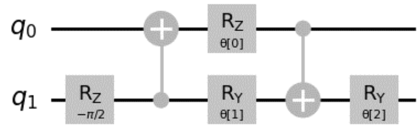

FIGURE 8. Pooling Layer

Overall, the following are all the stages carried out in this research, which can be seen in the following image:

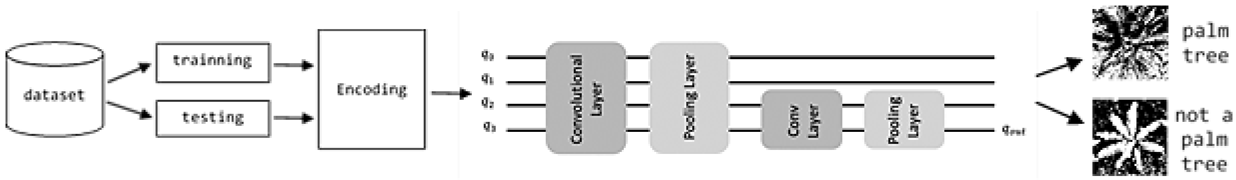

FIGURE 9. All stages of the research

RESULTS AND DISCUSSION

The entire process in this research uses Qiskit on IBM Quantum and is assisted by Dell inspiron 3881 hardware with 16 GB RAM memory (DDR4 SDRAM), Intel(R) Core(TM) i7-10700F CPU @ 2.90 GHz (8 Core) Processor and System Windows 10 Home Single Language 64 Bit Operation. Following are the complete results of the experiments carried out.

TABLE 1. Encoding ZFeatureMap

optimization methods	Accuracy (%)	
	training	testing
ADAM [learning rate: 0.001; iteration: 1000; beta_1=0.9; beta_2=0.99]	78.76	79.57
COBYLA [iteration: 1000]	89.21	91.23
AQGD [iteration: 1000; momentum=0.25]	88.27	89.31
NFT	88.78	88.85

TABLE 2. Encoding ZZFeatureMap

optimization methods	Accuracy (%)	
	training	testing
ADAM [learning rate: 0.001; iteration: 1000; beta_1=0.9; beta_2=0.99]	82.87	84.39
COBYLA [iteration: 1000]	90.32	91.89
AQGD [iteration: 1000; momentum=0.25]	90.16	90.24
NFT	90.31	90.45

TABLE 3. Encoding EfficientSU2

optimization methods	Accuracy (%)	
	training	testing
ADAM [learning rate: 0.001; iteration: 1000; beta_1=0.9; beta_2=0.99]	91.98	91.11
COBYLA [iteration: 1000]	94.70	92.50
AQGD [iteration: 1000; momentum=0.25]	92.17	91.74
NFT	92.89	91.55

From the experimental results we can see that the combination of the encoding method and the optimization method gives varying accuracy results, including: In the table with ZFeatureMap, the COBYLA optimization method gives the highest accuracy results in testing with 91.23%, while ADAM is lower with 79.57%. ZZFeatureMap provides significant accuracy improvements over almost all optimization methods, with COBYLA reaching 91.89%. The EfficientSU2 encoding method shows the highest results with COBYLA optimization achieving training accuracy of 94.70% and testing accuracy of 92.50%.

The encoding method in quantum models functions to convert classical data (such as image or numerical features) into a form that can be processed by a quantum system. Each encoding method has different characteristics in how they map data into quantum space. In ZFeatureMap: This encoding projects classical data into quantum Hilbert space using rotation about the Z axis of the qubit. This approach is suitable for data with simple patterns because it relies on amplitude-based representations, but may not be powerful enough to handle higher data complexity or more complicated non-linear relationships between features. In ZZFeatureMap: This encoding incorporates interactions between qubits (for example, interactions between two qubits via entanglement operations). This allows modeling more complex relationships among data features, resulting in the ability to handle data with more complex patterns. That's why this method produces higher accuracy compared to ZFeatureMap in most scenarios. In EfficientSU2: This method uses more complex unitary transformations, which allows encoding information in a more efficient and flexible way. With EfficientSU2, quantum systems can better capture variations in features and patterns in data, often leading to superior performance. This is also the reason why the highest accuracy in experimental results is produced by EfficientSU2 encoding. The complexity of the features that can be captured by different encoding methods. Methods that involve more inter-qubit interactions, such as ZZFeatureMap and EfficientSU2, are better at capturing complex patterns in the data, which in turn improves model accuracy.

ADAM (Adaptive Moment Estimation) is a popular optimization method due to its ability to adaptively adjust the learning rate based on the first (mean) and second (variance) moments of the gradient. However, in the context of QCNN, ADAM can sometimes get stuck in local minima, especially when working with quantum data that have more complex loss functions. This explains why ADAM does not always produce the highest accuracy, especially on more complex encodings like EfficientSU2. COBYLA (Constrained Optimization By Linear Approximations) is a constraint-based optimization method that does not use gradients, so it is more robust in dealing with optimization fields that are not smooth. This makes COBYLA able to avoid local minima traps and find better solutions in complex parameter spaces, especially in the case of complex encodings such as ZZFeatureMap and EfficientSU2. Therefore, this method often produces the highest accuracy in QCNN testing. Adaptive Quantum Gradient Descent (AQGD) adds momentum to the gradient, which helps prevent oscillations and accelerates convergence. Even though the performance is quite stable, momentum that is too high or low can cause the model to converge too slowly or actually miss the optimal solution, which can explain why AQGD does not always provide the highest accuracy. NFT is another non-gradient optimization method often used in quantum optimization. Although quite good at finding optimal solutions, these methods can be slow in some situations or tend to converge to non-optimal solutions if the search space is too large or complex.

Some optimization methods, such as COBYLA, excel at handling the more complex and non-smooth loss fields that often arise in quantum modeling. This method not only relies on gradient information but also uses a more flexible linear approximation approach, making it more effective in finding global solutions than methods such as ADAM which tend to rely more on gradients and can get trapped in local minima. The interaction between the encoding method and the optimization method also has a significant effect on the accuracy results. More complex encodings such as EfficientSU2 or ZZFeatureMap map data into a quantum space with larger dimensions and are more interactive, so the optimization field is also more complicated. Therefore, more robust optimization methods such as COBYLA tend to perform better on more complex encodings because they are able to handle greater terrain variations without relying on gradients. In contrast, simpler encoding methods such as ZFeatureMap have an easier field of optimization, whereas gradient methods such as ADAM may be quite efficient, although the results are not as good as those with more complex encodings.

CONCLUSIONS

The varying accuracy results in each experiment are caused by the complexity of the data captured by the encoding method and the effectiveness of the optimization method in finding the optimal solution. More sophisticated encodings are able to capture more complex patterns in the data, while optimization methods that are more robust in dealing with complex parameter spaces tend to provide higher accuracy.

ACKNOWLEDGMENTS

This research was funded by Bengkalis State Polytechnic in the 2024 Internal PNBP Grant Research Scheme.

REFERENCES

[1] https://bengkaliskab.bps.go.id/id/publication/2024/09/26/faa4841645a0d6ffc5b3370a/kecamatan-bengkalis-dalam-angka-2024.html
[2] Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. Nature Physics, 15(12), 1273-1278.
[3] Umeano, C., Paine, A. E., Elfving, V. E., & Kyriienko, O. (2023). What can we learn from quantum convolutional neural networks?. arXiv preprint arXiv:2308.16664.
[4] Meedinti, G. N., Srirekha, K. S., & Delhibabu, R. (2023). A Quantum Convolutional Neural Network Approach for Object Detection and Classification. arXiv preprint arXiv:2307.08204.
[5] Wei, S., Chen, Y., Zhou, Z., & Long, G. (2022). A quantum convolutional neural network on NISQ devices. AAPPS Bulletin, 32, 1-11.
[6] Oh, S., Choi, J., & Kim, J. (2020, October). A tutorial on quantum convolutional neural networks (QCNN). In 2020 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 236-239). IEEE.
[7] Hur, T., Kim, L., & Park, D. K. (2022). Quantum convolutional neural network for classical data classification. Quantum Machine Intelligence, 4(1), 3.
[8] Henderson, M., Shakya, S., Pradhan, S., & Cook, T. (2020). Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence, 2(1), 2.
[9] Simen, A., Flores-Garrigos, C., Hegade, N. N., Montalban, I., Vives-Gilabert, Y., Michon, E., ... & Martín-Guerrero, J. D. (2024). Digital-analog quantum convolutional neural networks for image classification. arXiv preprint arXiv:2405.00548.
[10] https://qiskit-community.github.io/qiskit-machine-learning/tutorials/11_quantum_convolutional_neural_networks.html
[11] Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[12] Powell, M. J. (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation (pp. 51-67). Springer Netherlands.
[13] Powell, M. J. (1998). Direct search algorithms for optimization calculations. Acta numerica, 7, 287-336.
[14] Powell, M. J. (2007). A view of algorithms for optimization without derivatives. Mathematics Today-Bulletin of the Institute of Mathematics and its Applications, 43(5), 170-174.
[15] Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., & Killoran, N. (2019). Evaluating analytic gradients on quantum hardware. Physical Review A, 99(3), 032331.
[16] Nakanishi, K. M., Fujii, K., & Todo, S. (2020). Sequential minimal optimization for quantum-classical hybrid algorithms. Physical Review Research, 2(4), 043158.