QRISE 2024 - IBM Quantum Challenge

Leveraging Dynamic Circuits in Quantum Convolution Neural Networks for Lung Cancer Classification

Team Name: QKul

In our study, we investigate the application of dynamic circuits in the classification of lung cancer CT scans of the IQ-OTHNCCD dataset. Our approach involves harnessing the power of Quantum Convolutional Neural Networks (QCNNs) to enhance accuracy and robustness. The dataset, a benchmark in lung tumor research, comprises diverse CT scan slices representing normal, benign, and malignant cases. By integrating dynamic circuits into the QCNN framework, we aim to improve lung cancer detection and contribute to the advancement of medical image analysis. Our results demonstrate that the integration of dynamic circuits with the QCNN framework significantly improves the prediction accuracy of lung cancer classification.

1. Quantum Convolutional Neural Networks with Dynamic Circuits

1.1 Convolutional Neural Network (CNN): The advent of CNNs has significantly advanced the field of image processing and pattern recognition. The hierarchical structure of CNN comprises convolutional layers, pooling layers, and fully connected layers that have the ability to learn intricate features from raw pixel data.

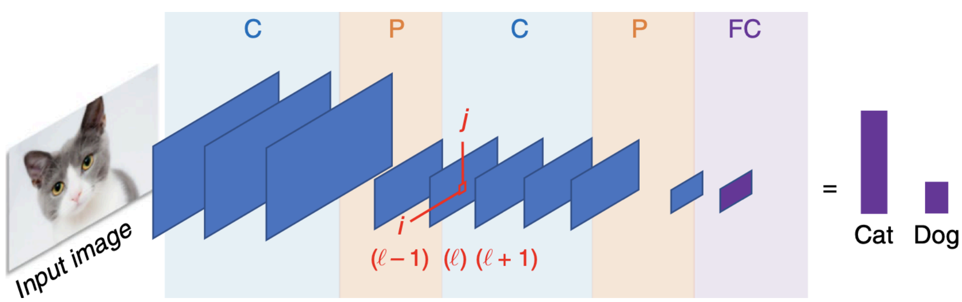


Fig. 1: Classification of cat and dog images using Classic CNN [1]

QCNNs emerge as a promising quantum counterpart to classical CNNs. QCNNs also employ convolutional layers to extract local features from input data. The distinguishing factor lies in their operation on quantum states, enabling them to encode and process information in a fundamentally different manner. In QCNNs, quantum gates replace classical convolutional filters, manipulating quantum states and allowing for parallel computation and potential speed-up in certain tasks.

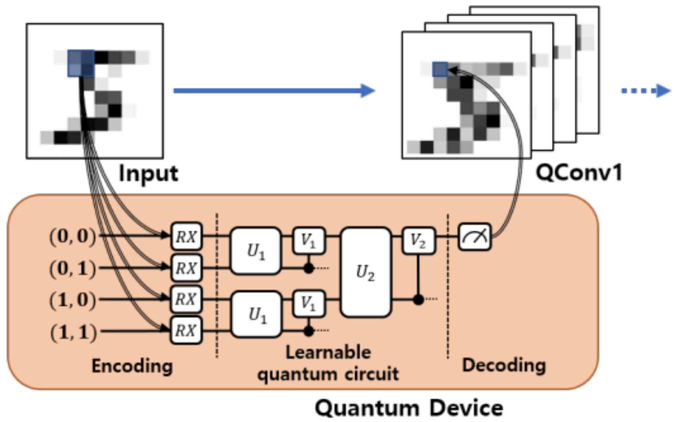


Fig. 2: Quantum convolution layer for image classification [2]

1.2 Dynamic Circuits: A quantum circuit is a sequence of quantum operations - including gates, measurements, and resets - acting on qubits. None of these operations in static circuits rely on run-time data. Static circuits might only have measurement activities at the very end. Conversely, dynamic circuits integrate classical processing during the qubit's coherence time. As a result, dynamic circuits can perform feed-forward operations and leverage mid-circuit measurements, utilizing the values obtained from the measurements to decide which operations to apply next. Additionally, for certain highly entangled quantum states, dynamic circuits are provably superior to static circuits in terms of preparing them for quantum simulation. [3]

When there are no experimental limitations, static and dynamic circuits have the same computational power. That is, any problem that can be represented with a dynamic circuit can also be represented as a static circuit. Dynamic circuits, however, can help overcome some of the drawbacks of actual hardware. Dynamic circuits, in particular, present novel possibilities for balancing circuit width and depth, which can make the difference between a working circuit and one whose output is indistinguishable from noise.

Qiskit offers four control flow constructs for classical feed-forward that are each implemented as a 'QuantumCircuit' method. [4] The constructs and their corresponding methods are:

- If statement: QuantumCircuit.if_test
- Switch statement: QuantumCircuit.switch
- For loop: QuantumCircuit.for_loop
- While loop: QuantumCircuit.while_loop

1.3 Implementation of Dynamic Circuits with QCNN: Our approach for the static circuit starts with encoding classical data into quantum data. Qubits are subjected to rotation gates, RX(θ), that correspond to pixel data. The learnable quantum circuit then implements the filter capable of uncovering the hidden state from the given input state. The decoding process relies on the measurement of one or more quantum states. By measuring these quantum states, classical data can be determined. This process is repeated by the learnable quantum circuit to ensure the hidden state is accurately determined.

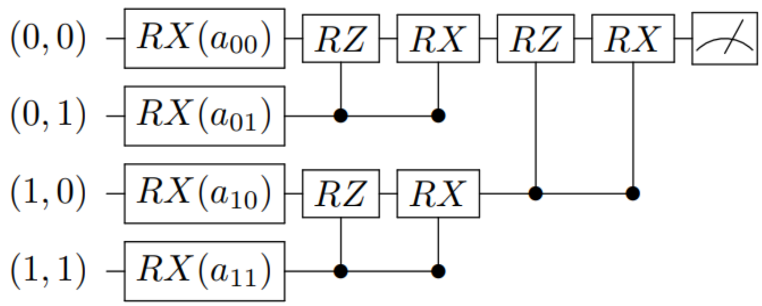
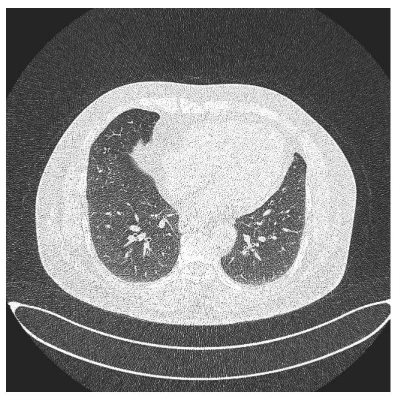
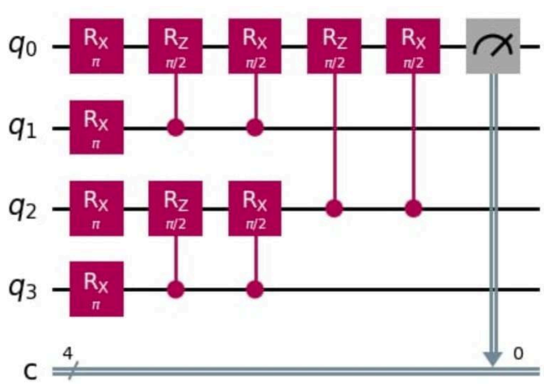
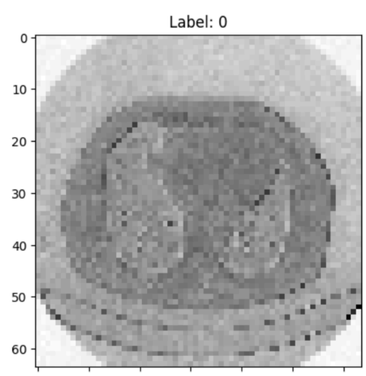
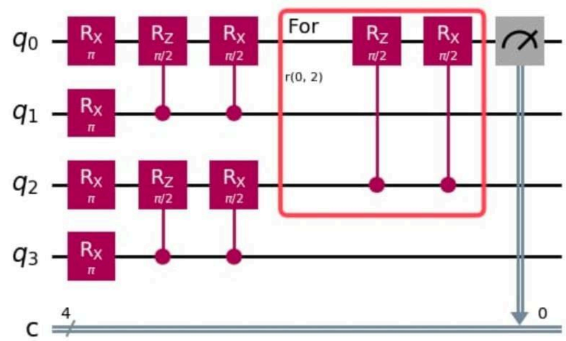


Fig. 3: Circuit architecture [2]

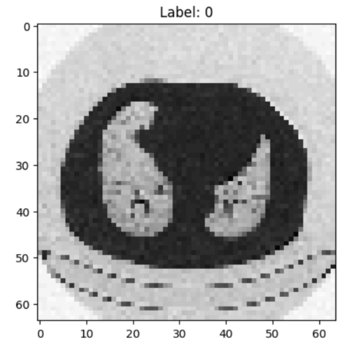
In our dynamic circuit methodology, we examine four distinct filters. These filters adhere to the same circuit architecture depicted in Fig. 3. We apply for-loops across various qubits to render the learnable quantum circuit dynamic.

Table 1: Static & dynamic filters and their respective results

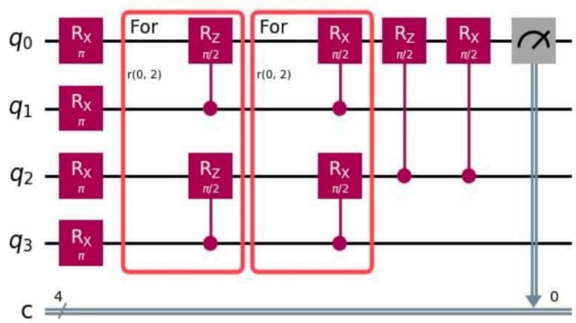
<table>
<tr><td colspan="2">
Original image</td></tr>
<tr><td>
Static filter</td><td>
Results of static filter</td></tr>
</table>



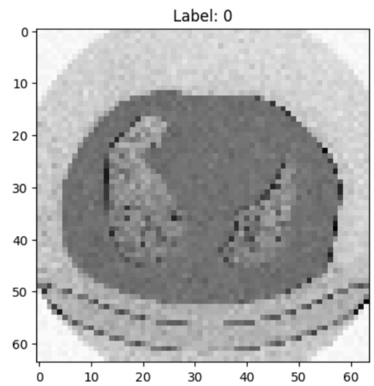
Dynamic filter 1



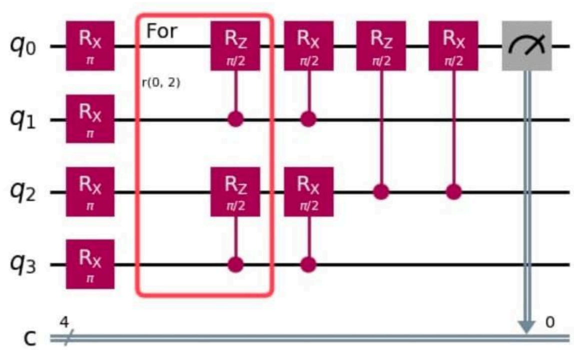
Results of dynamic filter 1



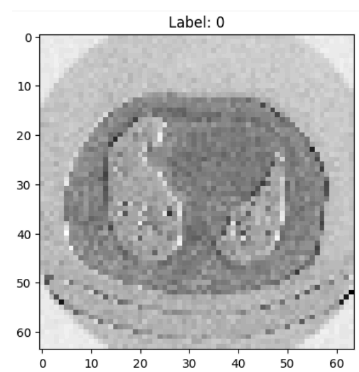
Dynamic filter 2



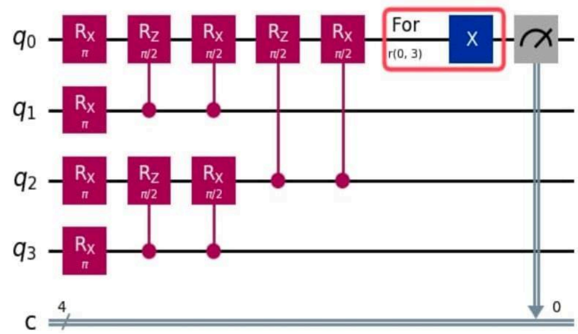
Results of dynamic filter 2



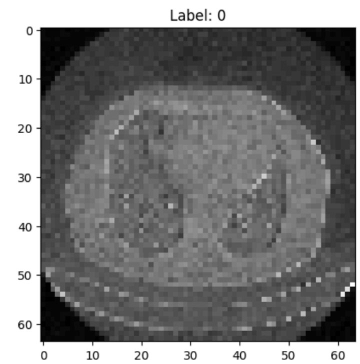
Dynamic filter 3



Results of dynamic filter 3



Dynamic filter 4



Results of dynamic filter 4

1.3.1 Classical Model: Flatten Layer (flatten_7) is the first layer of the model. It transforms its multi-dimensional input into a one-dimensional array. It can take an input of any batch size (represented by None) and will output a 4096-dimensional vector.

Dense Layer (dense_7) is the second layer of the model. It's a fully connected layer, meaning each neuron in this layer is connected to every neuron in the previous layer. It can take an input of any batch size and outputs a 3-dimensional vector.

We use a hybrid classical - quantum model for making the predictions and after quantum image processing we will use the following classical architecture.

Model: "sequential_7"

Layer (type)	Output Shape	Param #
flatten_7 (Flatten)	(None, 4096)	0
dense_7 (Dense)	(None, 3)	12,291

Total params: 36,875 (144.05 KB)
Trainable params: 12,291 (48.01 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 24,584 (96.04 KB)

Fig. 4: Classical model structure

1.3.2 Quantum Model: We utilize the Qiskit SDK and Tensorflow library for the model's implementation with following constraints:

1. The filter size for the quantum convolution layer is limited to 2x2.
2. The IQ-OTHNCCD lung cancer dataset is downsized to 25%, working with 274 images out of 1096 images, in order to save time and speed up processing.
3. Each image in the dataset has dimensions of 64x64.
4. In each epoch, 6 random images are chosen from the pool of 274 for the learning process.

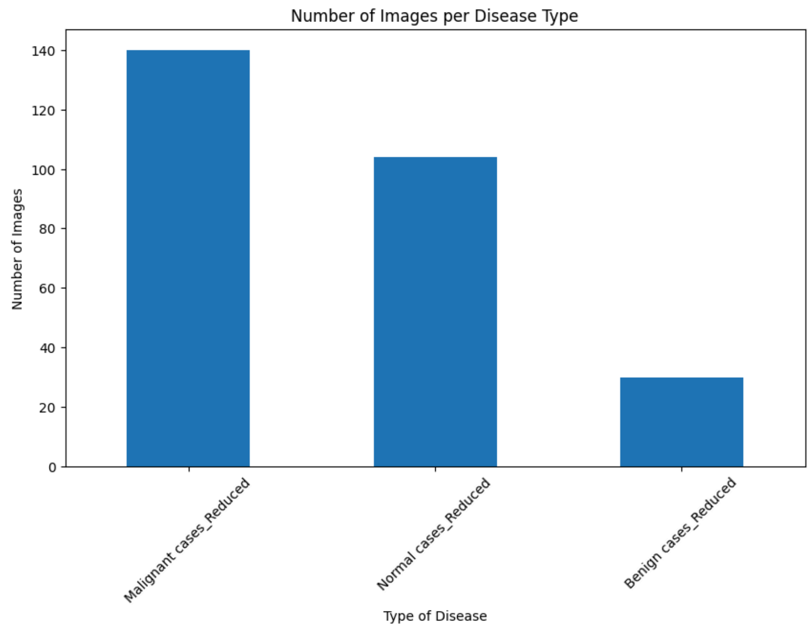


Fig. 5: Histogram representing type of disease

1.3.3 Results: The accuracy of each filter is as follows:

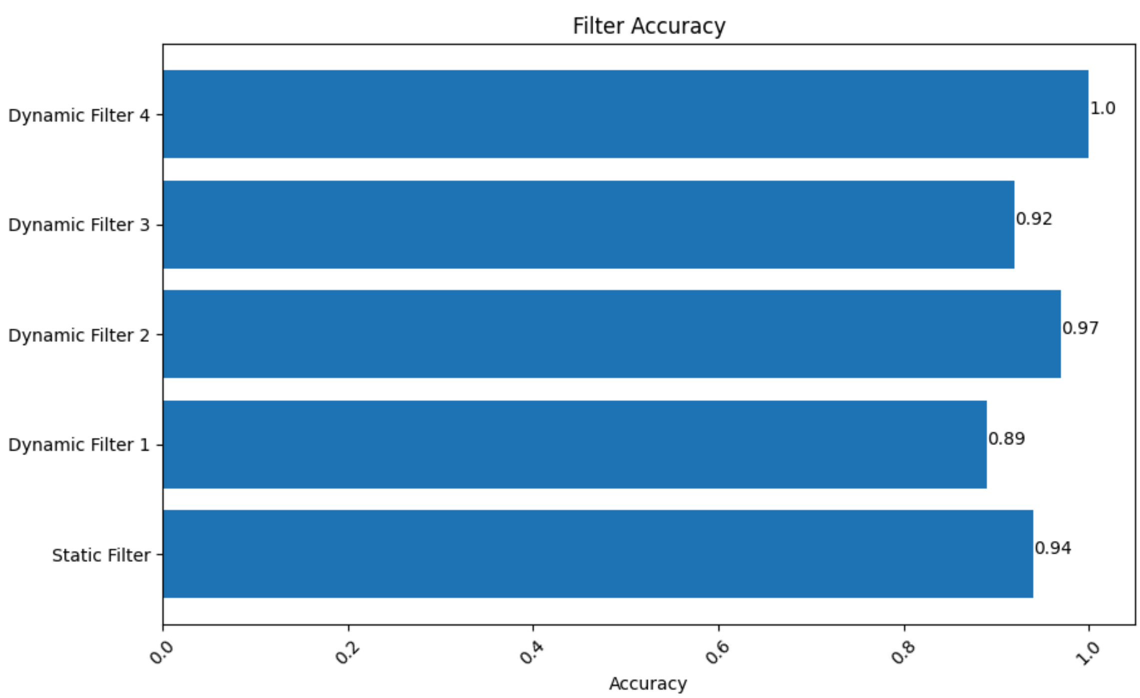


Fig. 6: Filter accuracy

Dynamic filter 4 shows an impressive 100% accuracy. It is established that dynamic filters 2 and 4 outperform the static filter by 3.19% and 6.38% respectively.

The accuracy and loss curves (training & validation); and confusion matrix for each filter are shown below:

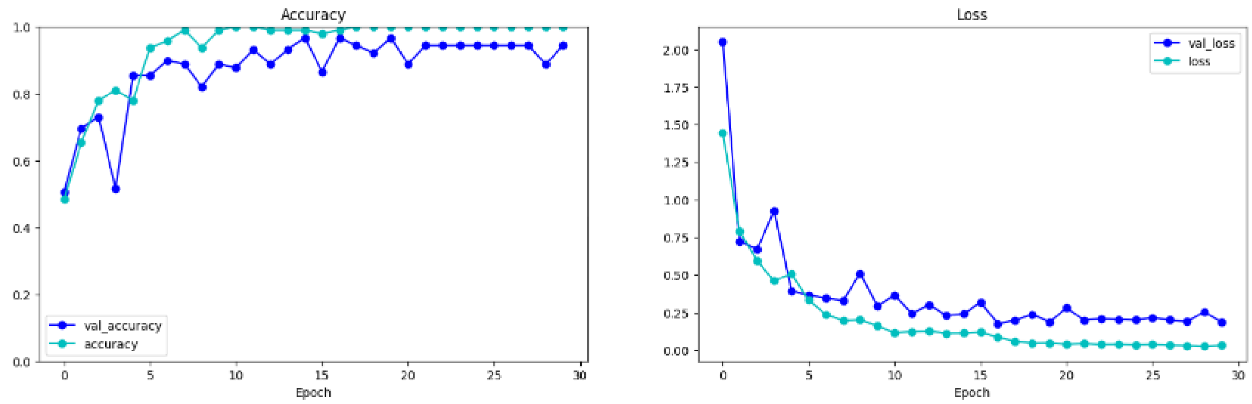


Fig. 7.1: Training process of static filter

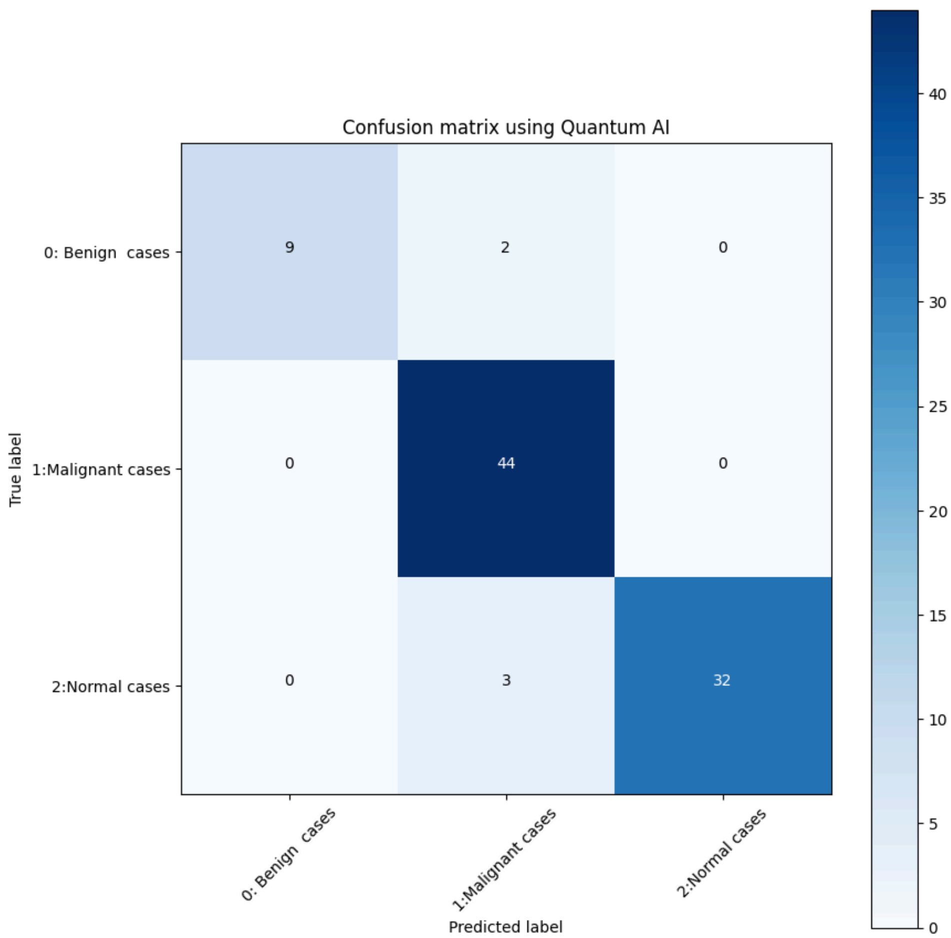


Fig. 7.2: Confusion matrix of static filter

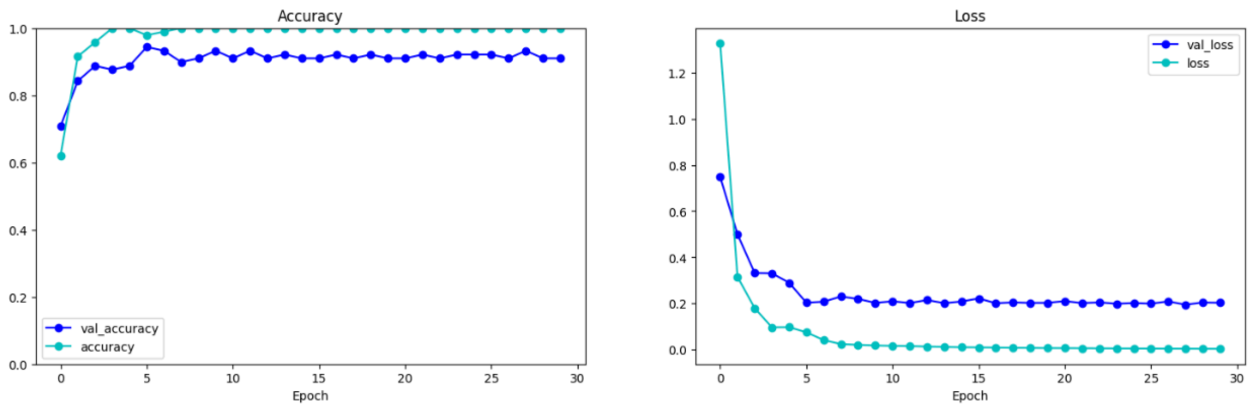


Fig. 8.1: Training process of dynamic filter 1

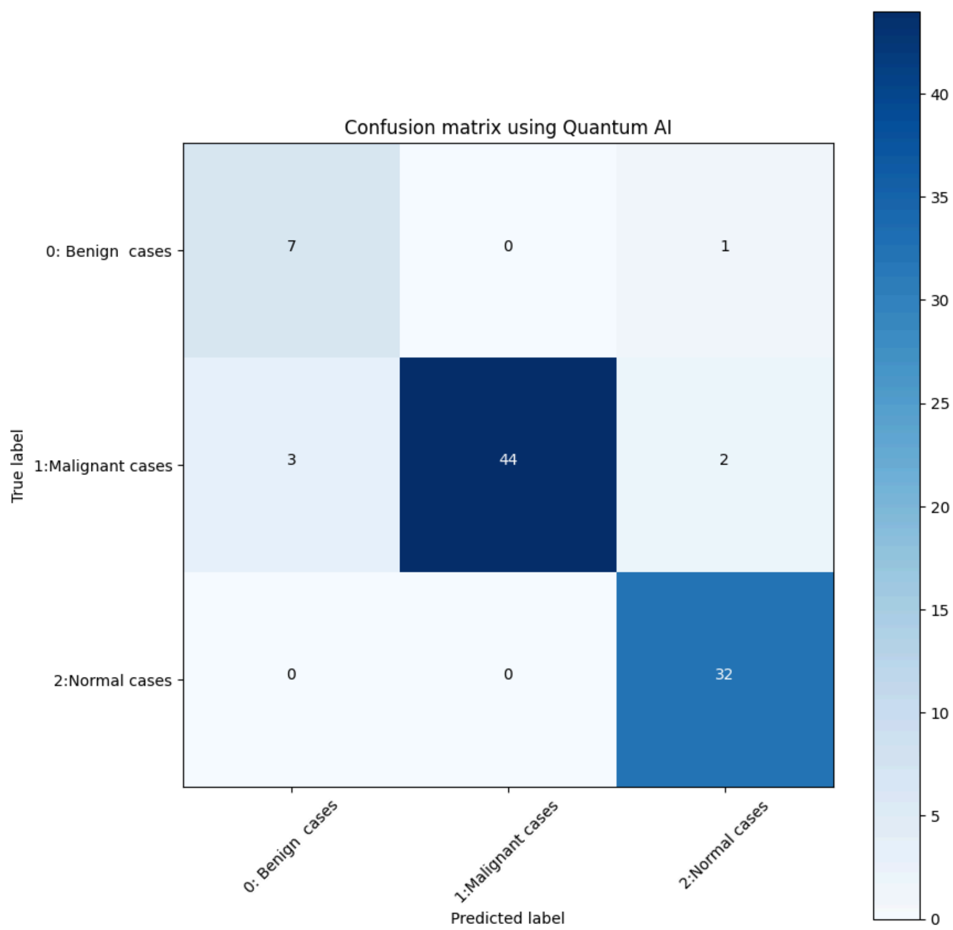


Fig. 8.2: Confusion matrix of dynamic filter 1

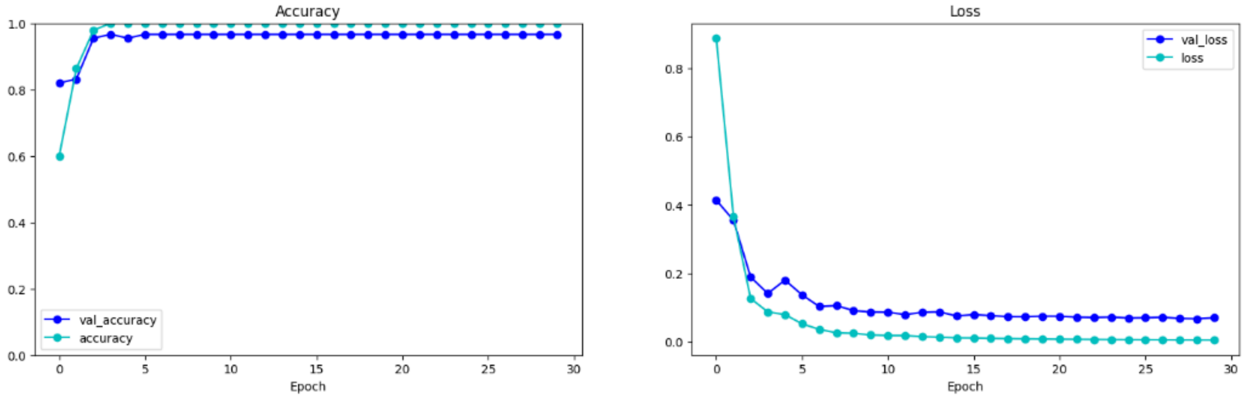


Fig. 9.1: Training process of dynamic filter 2

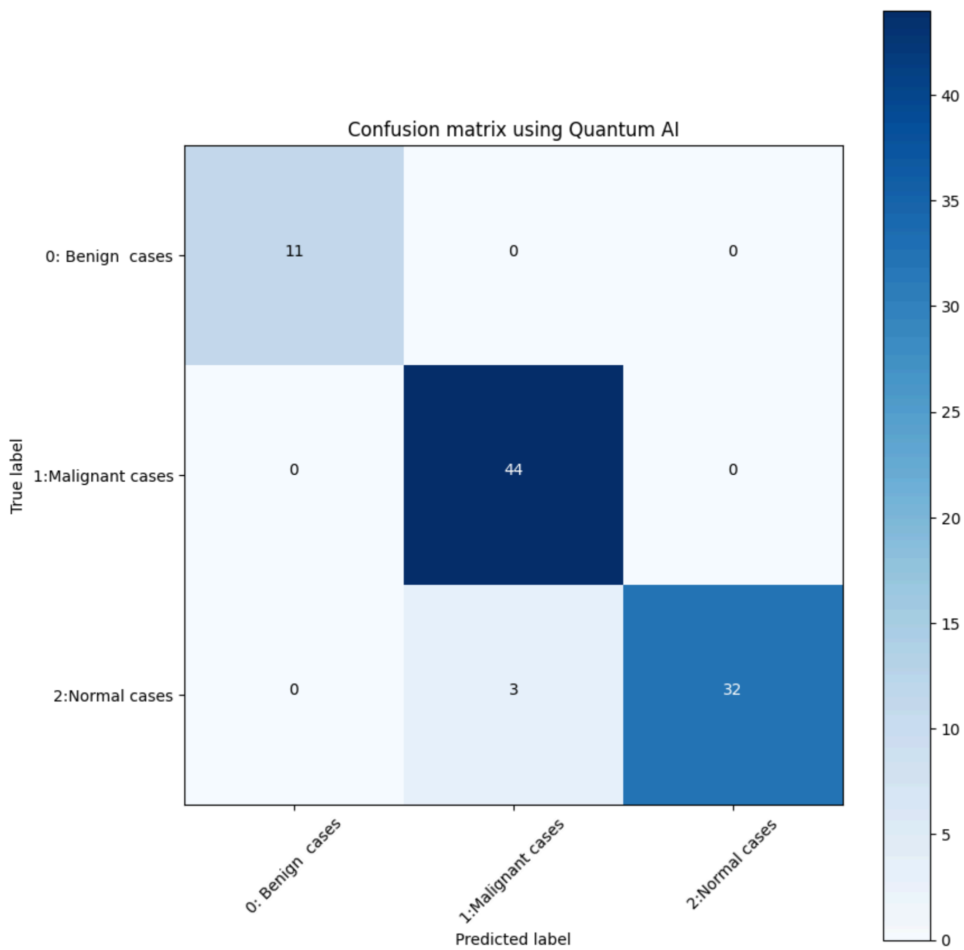


Fig. 9.2: Confusion matrix of dynamic filter 2

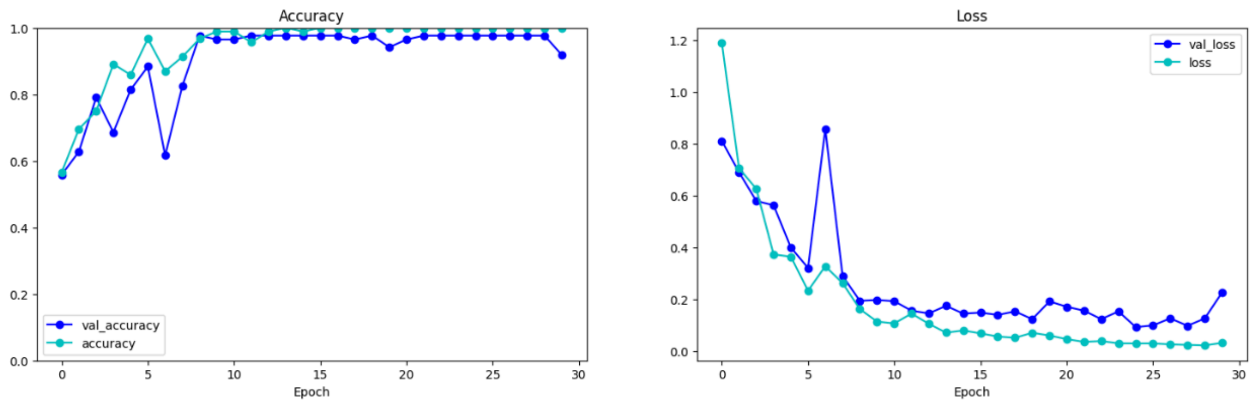


Fig. 10.1: Training process of dynamic filter 3

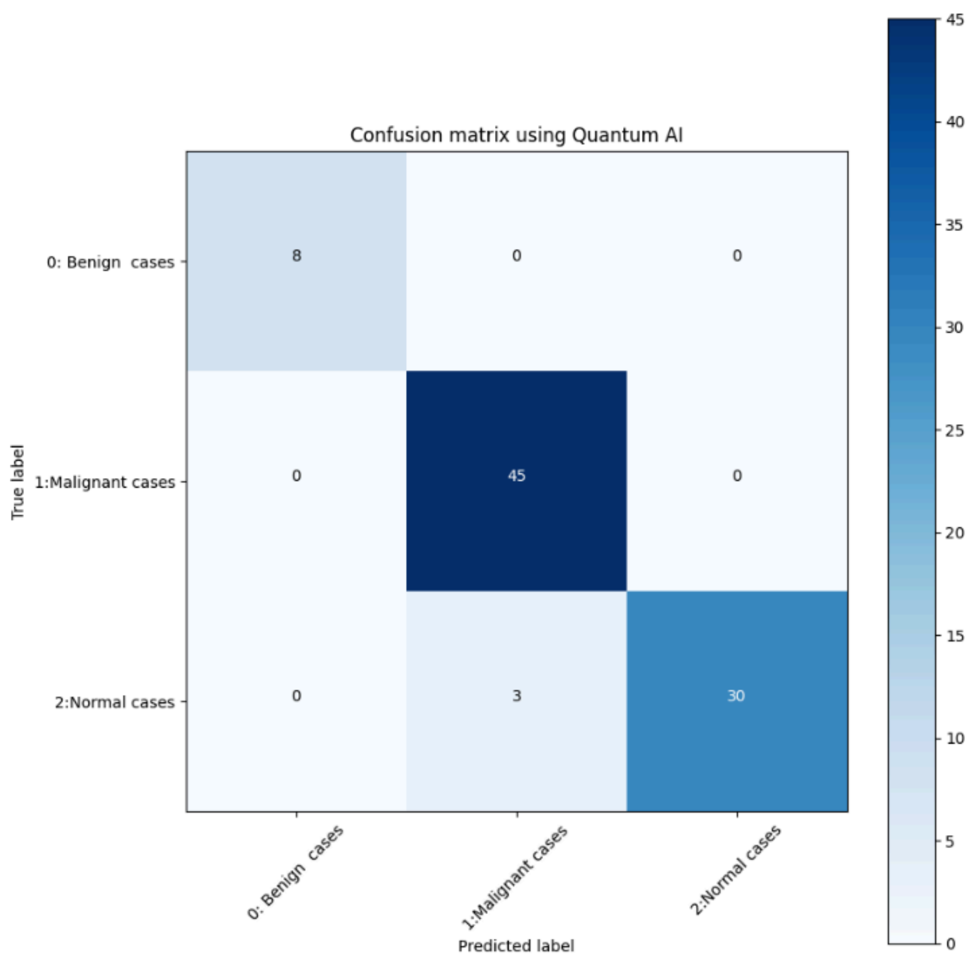


Fig. 10.2: Confusion matrix of dynamic filter 3

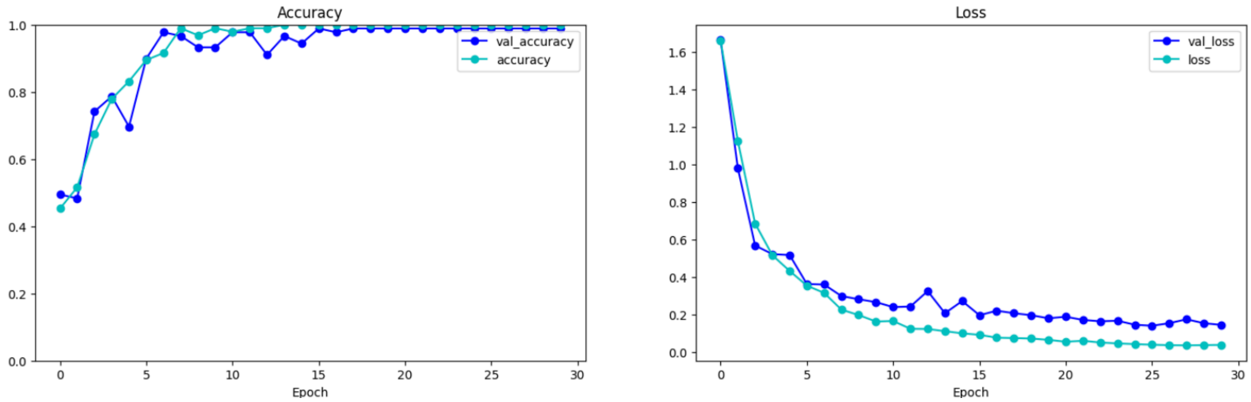


Fig. 11.1: Training process of dynamic filter 4

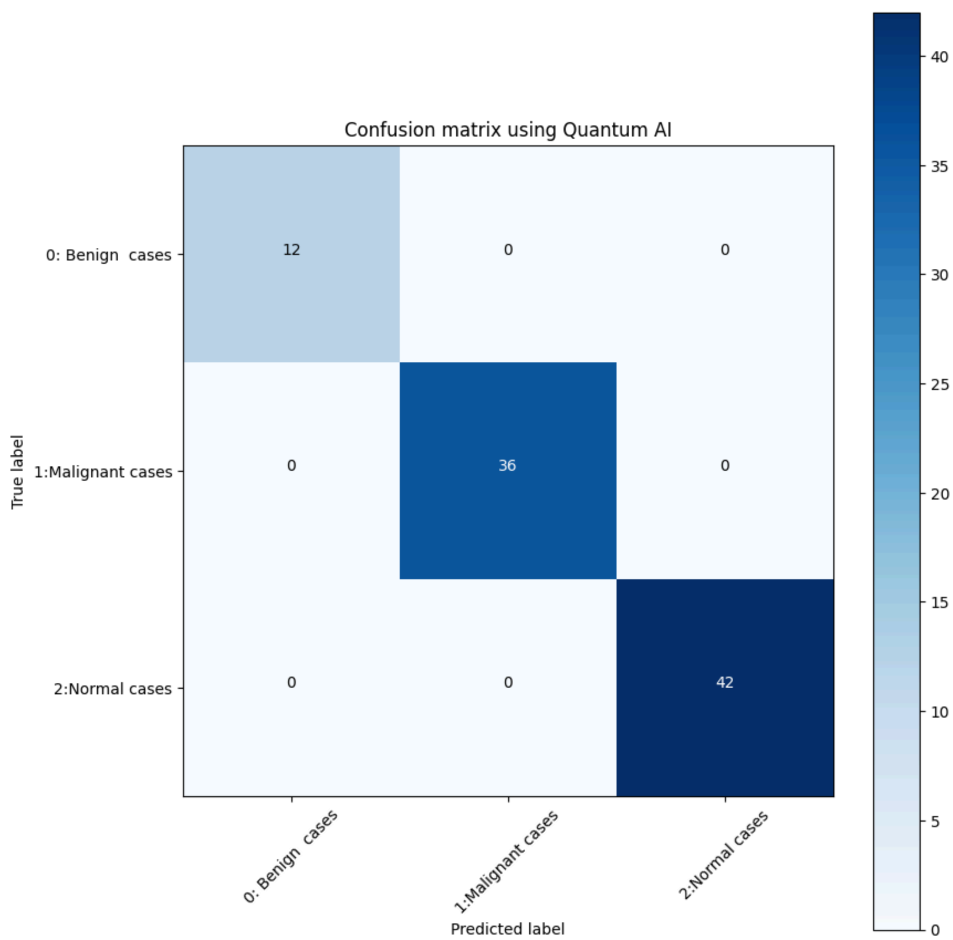


Fig. 11.2: Confusion matrix of dynamic filter 4

1.4 Noise:

The noisy model simulates the effects of depolarizing noise on a quantum computer using Qiskit. Depolarizing noise is a common error model that represents the type of errors that can occur due to imperfect isolation of qubits from their environment or non-ideal gate operations. The probability value 0.09 (or 9%) of gates indicates that each time a gate is applied, there is a 9% chance that the gate will cause a depolarizing error. The u1, u2, and u3 gates are specified to have depolarizing errors associated with them. This means that whenever these gates are used in a quantum circuit that is simulated with this noise model, the specified depolarizing errors will be applied to them.

```
NoiseModel:
  Basis gates: ['cx', 'id', 'rz', 'sx', 'u1', 'u2', 'u3']
  Instructions with noise: ['u3', 'u1', 'u2']
  All-qubits errors: ['u1', 'u2', 'u3']
```

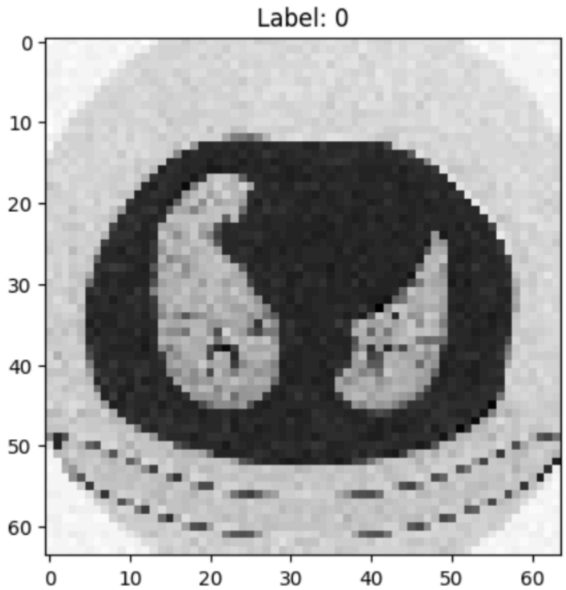
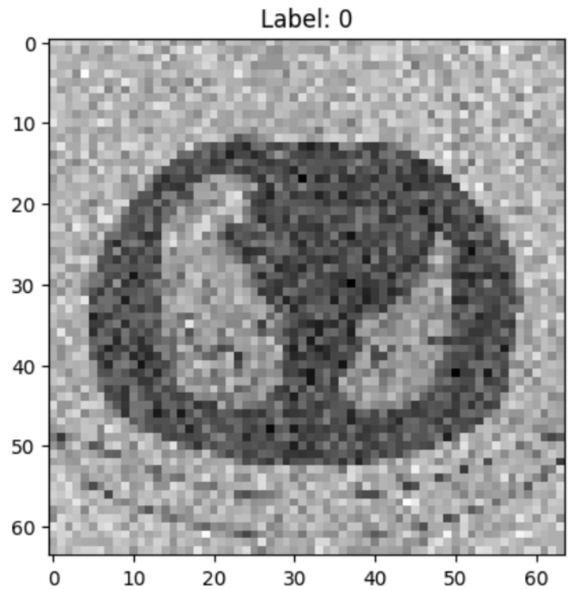
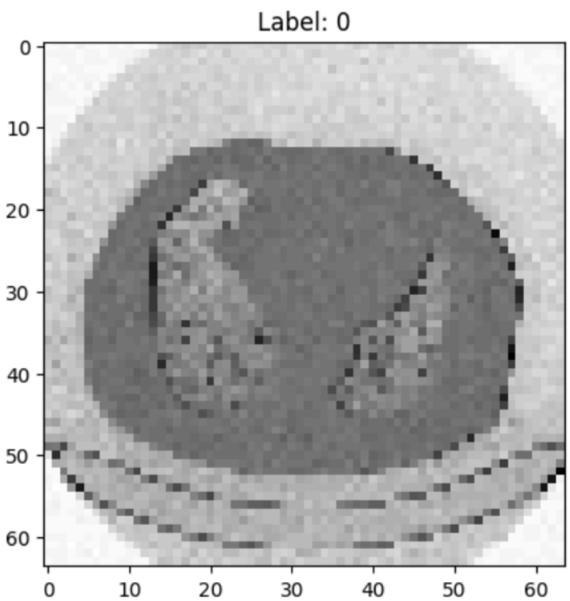
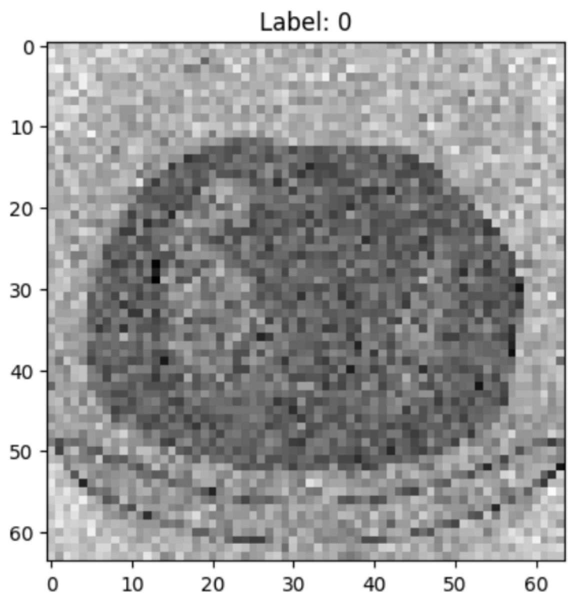
Fig. 12: Noise model structure

- U1 gate: A single-qubit gate that performs a rotation about the Z-axis of the Bloch sphere. The u1(λ) gate is a phase shift gate where λ is the angle of rotation.
- U2 gate: A single-qubit gate that performs a general rotation about the X and Y axes of the Bloch sphere. The u2(φ, λ) gate is essentially equivalent to applying a Hadamard gate followed by a u1 gate, and another Hadamard gate.
- U3 gate: The most general form of a single-qubit gate, which performs an arbitrary rotation about the X, Y, and Z axes. The u3(θ, φ, λ) gate can generate any single-qubit gate.

The reasons why depolarizing errors may occur are as follows:

1. **Imperfect gate operations:** Quantum gates, while ideally precise, can have operational errors due to factors like calibration errors and unintended environmental interactions. These can cause depolarizing errors, randomizing the qubit's state.
2. **Thermal noise:** Despite cooling quantum systems to minimize thermal noise, residual interactions can still cause depolarizing errors. These can flip the qubit state or cause phase shifts, degrading operation fidelity.
3. **Decoherence:** Quantum systems can lose their quantum properties through decoherence, a process caused by environmental interactions. This leads to the loss of superposition and entanglement, and randomizes quantum information over time.
4. **Electromagnetic interference:** Quantum systems are sensitive to electromagnetic fields, which can perturb qubit energy levels and lead to errors. Ambient electromagnetic noise can induce state transitions or disturb phase relationships, contributing to depolarizing errors.
5. **Cross-talk between qubits:** In quantum processors, close proximity of qubits can lead to cross-talk, where an operation on one qubit affects a neighboring one. This can cause depolarizing errors as qubit states become mixed due to these unintended interactions.

Table 2: Comparison of processed images using dynamic filters with and without noise

 **Image processed by dynamic filter 1 without noise**	 **Image processed by dynamic filter 1 with noise**
 **Image processed by dynamic filter 2 without noise**	 **Image processed by dynamic filter 2 with noise**

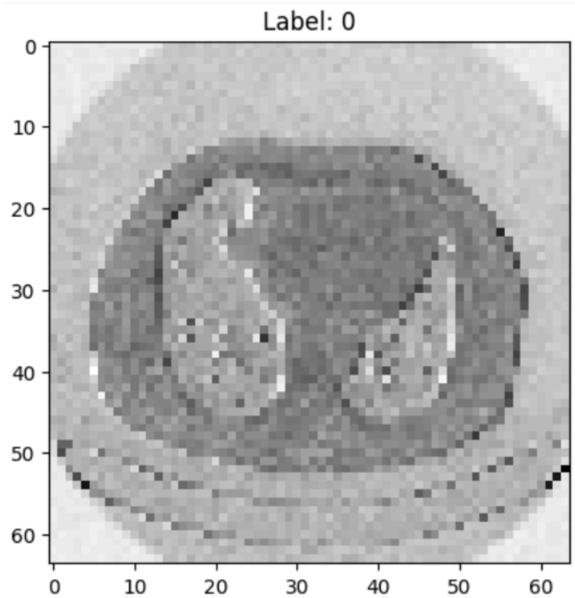


Image processed by dynamic filter 3 without noise

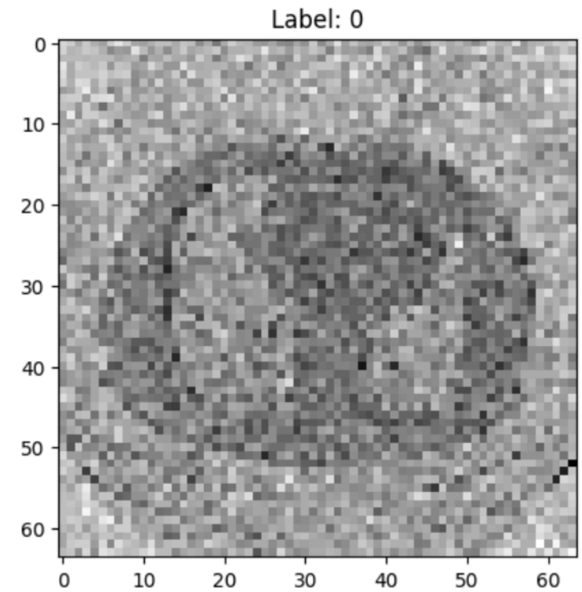


Image processed by dynamic filter 4 with noise

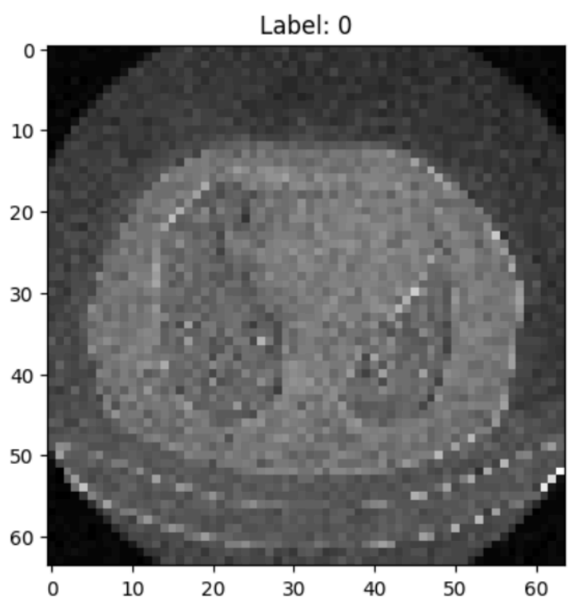


Image processed by dynamic filter 4 without noise

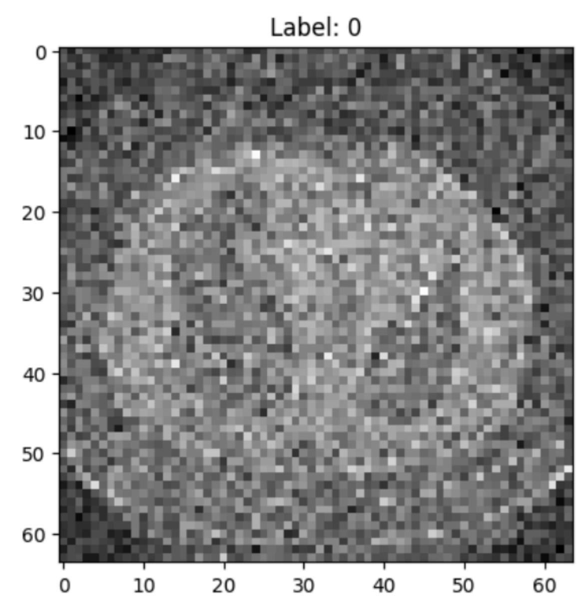


Image processed by dynamic filter 4 with noise

2. Conclusion

The study presents a significant advancement in the field of quantum computing and its application in medical diagnosis. It leverages dynamic circuits in Quantum Convolutional Neural Networks and evaluates their performance against a static filter for lung cancer classification.

- Dynamic filter 2 and 4 outperform the static filter by approximately 3.19% and 6.38% respectively, demonstrating the potential of dynamic circuits in improving the accuracy of lung cancer classification.
- However, the performance of dynamic filter 1 and 3 indicates that not all dynamic circuits guarantee an improvement over static ones. This highlights the importance of careful design and selection of dynamic circuits in QCNN.

The study also introduces a noisy model that simulates the effects of depolarizing noise on a quantum computer using Qiskit. This model assumes a 9% chance of depolarizing error each time a gate is applied, and provides a realistic representation of the type of errors that can occur due to imperfect isolation of qubits from their environment or non-ideal gate operations. The inclusion of this noise model adds a layer of realism to the study, acknowledging the practical challenges faced in quantum computing.

3. Future Directions

- Optimizing the model for real-time applications by further optimization of the quantum circuits used in the model. This could involve exploring different types of quantum gates, different circuit architectures, or different methods for compiling the circuits.
- Expanding the dataset may improve the robustness and generalizability of the model. This could include images from different sources and different imaging techniques. Additionally, leveraging the parallel processing of images could allow the model to analyze multiple images simultaneously. This approach not only has the potential to significantly reduce the time required for training the model, but also makes it feasible to handle larger and more diverse datasets.
- Increasing the filter size could potentially capture more complex features in the images. Batch processing techniques may also speed up the learning process.

References

[1] Cong, I., Choi, S. & Lukin, M.D., “Quantum convolutional neural networks,” Nature Physics 15, pp. 1273–1278, 2019.
[2] S. Oh, J. Choi, and J. Kim, "A Tutorial on Quantum Convolutional Neural Networks (QCNN)," International Conference on Information and Communication Technology Convergence (ICTC), pp. 236-239, 2020.
[3] IBM, “Bringing the full power of dynamic circuits to Qiskit Runtime,” https://www.ibm.com/quantum/blog/quantum-dynamic-circuits
[4] IBM, “Classic feedforward and control flow,” https://docs.quantum.ibm.com/build/classical-feedforward-and-control-flow