Quantum-Based Binary Classification of Histological Images of Salivary Glands with Sjögren Syndrome

Cesar Esteban Díaz Medina
cediazme@unal.edu.co
Computer Systems Engineering
Universidad Nacional de Colombia

Diego Esteban Quintero Rey
diquintero@unal.edu.co
Computer Systems Engineering
Universidad Nacional de Colombia

ABSTRACT

This study compares the performance of two neural network architectures—quanvolutional and classical convolutional—via a series of experiments. The methodology includes dataset preprocessing, downscaled and standard versions, utilizing a defined quantum circuit for the quanvolutional layer. The training involves 10 splits with K-Fold cross-validation, exploring variations like MedViT feature extraction and data augmentation. Initial findings show QNN superiority, performance variations, and the need for further exploration, particularly in quanvolutional layer optimization for enhanced quantum architecture performance in image classification tasks. The video describing this work can be found at: https://www.youtube.com/watch?v=E93dtL7zJ2w

I. OBJECTIVE

The aim of this study is to evaluate and contrast the performance of two neural network architecture: one incorporating a quanvolutional layer and the other utilizing a classical convolutional layer. A series of experiments is conducted to accomplish this. Additionally, as an initial exploration, a variational quantum classifier employing transfer learning is tested.

II. INTRODUCTION

Sjögren's Syndrome

Sjögren's Syndrome (SS) presents itself as a chronic autoimmune disease characterized by inflammation of the exocrine glands, especially the salivary and lacrimal glands. This inflammation leads to symptoms of dryness in various mucous membranes, especially in the mouth and eyes, manifesting as xerostomia and dry keratoconjunctivitis [1, 2].

The clinical presentation of SS varies significantly and tends to be more prevalent in women, with an increase in dry symptoms as they age [3]. Glandular symptoms, such as dry eyes, are noticeable and can have a significant impact on the quality of life of patients [1-3].

Given the complexity and heterogeneity of this disease, it is crucial to analyze histological images of tissue related to SS using convolutional neural networks. This would enable a more precise and objective evaluation of pathological characteristics, providing a deeper understanding of the disease and assisting in its diagnosis and treatment [3].

The classic histopathological characterization of SS is achieved through the biopsy of salivary glands. It is defined as the presence of one or more foci containing dense aggregates of 50 or more lymphocytes, located periductally or perivascularly adjacent to acini with normal appearance. The presence of more than one lymphocytic focus is considered a diagnostic criterion for SS [7]. However, alterations in acinar architecture and ductal hyperplasia can also be observed, along with increased angiogenesis, accumulation of hyaline material in the lumen of atrophied ducts, and other findings including the formation of germinal centers, presence of interstitial fibrosis, and ductal dilation [8].

Classical Neural Networks in Digital Histopathology

In recent decades, the field of digital histopathology has undergone significant progress in image analysis [4]. In its early stages, in the 1960s, it relied on predefined features and simple algorithms for the processing of histopathological images [4]. However, one of the most notable advances occurred with the introduction of convolutional neural networks (CNN), specifically designed for computer vision tasks. These CNNs automate feature extraction and pattern learning, enabling computers to analyze histopathological images with precision and efficiency, without relying on predefined features and manual intermediate steps [4], [5].

One of the most challenging tasks is the classification of medical images, which has seen significant advancements with the development of Convolutional Neural Networks (CNNs) [9]. Despite successes in learning robust representations, CNNs struggle to capture long-term dependencies in medical visual data, which often exhibit variations in texture, shape, and organ size [10], [11]. In response to these limitations, Transformer architectures have emerged, using self-attention mechanisms to model long-term dependencies, and demonstrating advantages over CNNs in terms of performance [12], [13]. However, the application of Transformers in realistic clinical settings poses challenges due to the need for extensive training datasets and computational limitations [14].

Quanvolutional Neural Networks (QNN)

Quanvolutional Neural Networks (QNNs) enhance the capabilities of Convolutional Neural Networks (CNNs) by harnessing certain potentially powerful aspects of quantum computation. QNNs introduce a novel type of transformational layer, known as the quantum convolutional (or quanvolutional) layer, into the standard CNN architecture. These quanvolutional layers consist of a set of N quantum filters that function similarly to classical convolutional layers, generating feature maps by locally transforming input data. The key distinction lies in the fact that quanvolutional filters extract features from input data by transforming spatially-local subsections using random quantum circuits. The utilization of random nonlinear features is recognized for enhancing accuracy or reducing training times in various machine learning algorithms, as seen in CNNs with random convolutions and echo state networks.

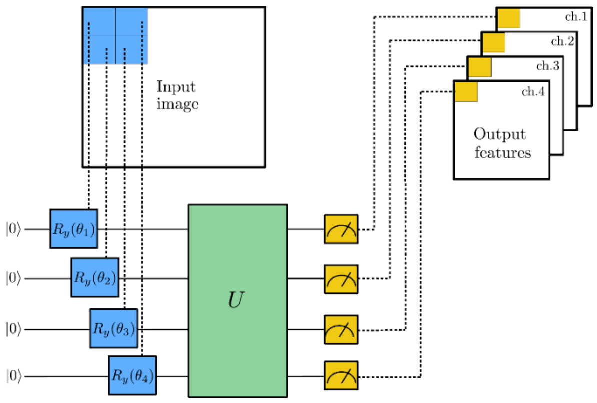


Figure 1 - Quanvolutional Layer

Several quantum variations of classical models have recently emerged, including Quantum Reservoir Computing (QRC), Quantum Circuit Learning (QCL), Continuous-Variable Quantum Neural Networks, Quantum Kitchen Sinks (QKS), Quantum Variational Classifiers, and Quantum Kernel Estimators. The QNN approach aims to leverage the unique aspects of quantum circuit transformations within a machine learning framework, differing from previous works in the methodology of processing classical information into and out of quantum circuits and the flexible integration of such computations into state-of-the-art deep neural network models.

Another distinctive aspect of the QNN approach is the absence of variational tuning for the quantum components of the model. QNNs are essentially an extension of classical CNNs, featuring an additional transformational layer called the quanvolutional layer. Quanvolutional layers integrate similarly to classical convolutional layers, allowing users to define arbitrary integer values for the number of quanvolutional filters, stack new quanvolutional layers on top of any other layer in the network stack, and specify layer-specific configurational attributes.

Quanvolutional filters, satisfying these conditions, are designed to be highly generalizable and easy to implement in any architecture, similar to their classical counterparts. The number, order, and specific parameters of such layers are entirely customizable based on the end user's specifications. Quanvolutional filters produce feature maps by applying them to an input tensor, transforming spatially-local subsections using the quanvolutional filter. Unlike classical convolutional filters that perform a simple element-wise matrix multiplication operation, quanvolutional filters employ a quantum circuit for input data transformation, which can be structured or random.

Variational Quantum Classifiers

The variational quantum classifier represents a variational algorithm where researchers interpret the measured expectation value as the output of the classifier. This algorithm was introduced by multiple groups in 2018. For a binary classification problem with input data vectors x_i and binary output labels $y_i = \{0, 1\}$, a parameterized quantum circuit is constructed for each input data vector. This circuit outputs the quantum state $|\psi(x_i, \theta)\rangle = U_{W(\theta)} U_{\phi(x_i)} |0\rangle$, where the variational circuit unitary is denoted by $U_{W(\theta)}$ and the data encoding circuit unitary is denoted by $U_{\phi(x_i)}$. After creating and measuring the circuit of qubits, a bitstring of length n is obtained, and a binary output, determined by a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, is derived from this bitstring. The parity function, a popular choice, returns True if the input bit string has an odd number of 1s and False otherwise.

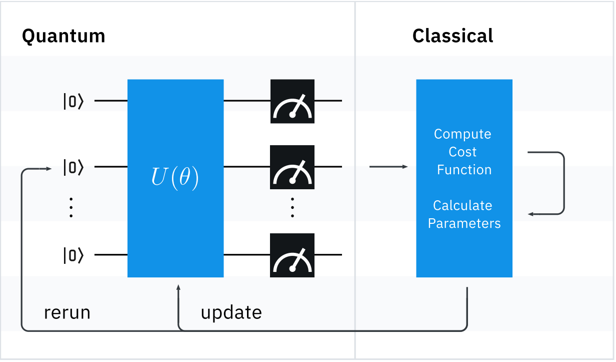


Figure 2 - Variational Quantum Circuit (VQC)

During the training phase, efforts are made to find the values for θ that yield the best predictions. The classical computer compares the predicted labels $\hat{y}_i$ to the provided labels y_i, and the success of the predictions is calculated using a cost function. Based on this cost, the classical computer utilizes a classical optimization algorithm to determine another value for θ. This new θ is then used to run a new circuit, and the process is repeated until the cost function stabilizes.

As with all variational algorithms, determining the optimal parameters of the variational circuit is computationally intensive and depends on the optimization method used. The optimal circuit parameters, denoted as θ^*, are found when the minimum of the loss function, $L(\theta)$, is reached. However, establishing a simple relationship between the loss function and the circuit parameters is challenging. The loss landscape is complex. Optimization methods navigate this landscape in search of the minimum, illustrated by the black points and lines. Notably, two-thirds of the searches often converge to a local landscape minimum rather than a global one.

Optimization methods are generally categorized into two groups: gradient-based and gradient-free methods. Gradient-based methods identify an extreme point where the gradient is zero, and a search direction is determined by the derivative of the loss function. These methods can have slow convergence and no guarantee of reaching the optimal solution. Gradient-free methods, on the other hand, are useful when derivative information is unavailable or impractical to obtain. These methods are robust in finding global optima but require higher computational capacities, especially for problems with high-dimensional search spaces.

Regardless of the optimization method used, if the loss landscape is flat, determining the direction to search can be challenging. This situation is known as a barren plateau. For a wide class of reasonable parameterized quantum circuits, the probability that the gradient along any reasonable direction is non-zero to some fixed precision is exponentially small as a function of the number of qubits. To address this problem, one approach is to use structured initial guesses, as seen in quantum simulation. Another possibility is to view the full quantum circuit as a sequence of shallow blocks, selecting some parameters randomly and determining the rest of the parameters such that all shallow blocks implement the identity to restrict the effective depth. This is currently an area of active investigation.

III. DATASET

More than 80 years ago, the National Cancer Institute was established as a prominent center for researching various types of cancer in Colombia and Latin America. This institution not only provides specialized services for the diagnosis and clinical analysis of cancer patients but also manages an extensive dataset linked to histological slides displaying various pathologies.

The current project is based on salivary gland biopsies previously diagnosed by a group of expert pathologists. This sample consists of 200 histological images per class, both with and without Sjögren's syndrome (SS). The images are balanced with 50% healthy tissue and 50% affected tissue. Permissions granted by the pathology department for the use of this data restrict the project to an analysis and implementation of artificial intelligence tools. However, publication is limited without prior consultation and the completion of forms authorizing the use of personal data from patients whose samples were taken.

For data processing, photographs of histological slides from minor salivary gland biopsies were taken. These biopsies were prepared using the conventional protocol of cutting with a microtome into paraffin cassettes. Each slide contains a specific number of biopsies and patients. The images were captured using the Motic Digital Slide Assistant program, which allowed standardization of the image capture field, resolution, and focal distance of each image (100 μm).

	Plaques	**Patients**	**Images**	**Biopsies**
Negative	1	4	57	3/1/2/2
	2	3	68	2/3/2
	3	8	75	2/2/2/2
Positive	4	4	112	4/3/2/3
	5	8	88	2/2/2/2
Total	5	19	400	43

Figure 3 - Dataset Collection

IV. METHODOLOGY

This study employs a specific methodology comprising several steps. Initially, the dataset undergoes preprocessing to create two distinct sets. The first set is a downscaled variant achieved through a sequence of transformations. The images undergo conversion to grayscale, followed by histogram equalization, binary thresholding at two levels within the graylevel range $(1, 10)$, and eventual resizing to dimensions of 28×28. The inspiration for this downsampling approach originates from [15], where the authors utilized a similar technique for classification in Quantum Machine Learning (QML) models. In our case, the primary objectives are twofold: firstly, to diminish the computational time required for feature extraction from the quanvolutional layer, and secondly, to investigate whether binary thresholding alone is sufficient for classifying images by detecting dark spots (agglomerations of lymphocytes). This approach is justified by the observation that non-syndrome images generally exhibit a lower degree of lymphocyte agglomeration. The second version of the dataset is identical to the first, excluding the binary thresholding step, and resized to dimensions of 32×32. OpenCV library is employed for all the preprocessing tasks, and the resulting preprocessed datasets are saved as HDF5 files.

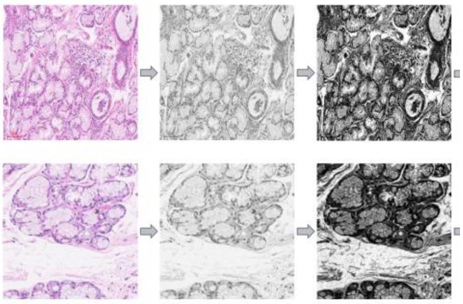


Figure 4 - Histogram Equalization Pre-processing

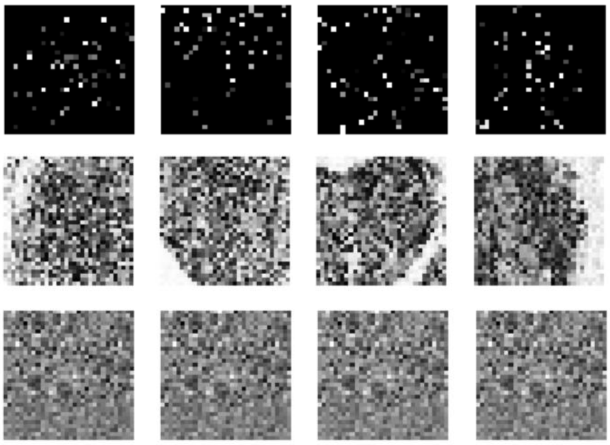


Figure 5 - Downsampled (top), Original (center), MedViT Features (down) examples

The subsequent stage involves loading the HDF5 files containing pre-processed data into the environment. In this study, Google Colab serves as the platform for this task. Following data loading, a normalization process is implemented to ensure that the image values fall within the range of $(0, 1)$.

Subsequently, a quantum circuit is defined, which serves as the foundation for the quanvolutional layer responsible for feature extraction. This circuit comprises 4 qubits and is parameterized by a vector, ϕ, encompassing 4 components. Each component corresponds to a cell within a 2×2 convolutional filter applied to a localized region of the input image. The input components' values are encoded into quantum states using the RY gate, with the angle parameter calculated by multiplying the input component by π. This layer of the circuit represents the quantum encoding step. Following this, a layer of randomly chosen single qubit rotations and 2-qubit entangling gates, acting on randomly chosen qubits, is added to the circuit. Ultimately, the 4 rotated qubits are measured in the Z basis, constituting the outputs of the convolutional layer. Consequently, this convolutional layer possesses 1 input channel and yields 4 output channels. A stride of 2 is applied to this quanvolutional layer. Pennylane is employed for circuit implementation.

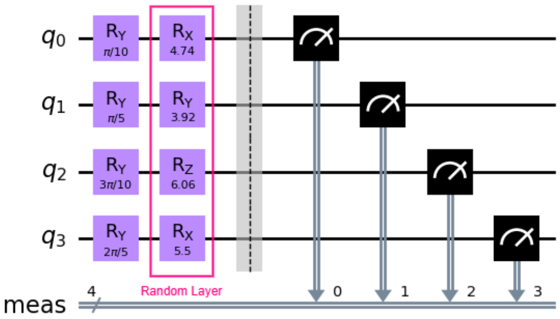


Figure 6 - Quanvolutional Layer Random Circuit

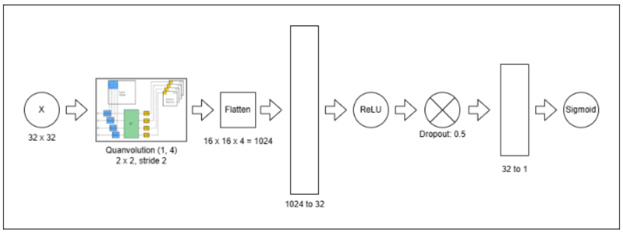


Figure 7 - QNN Architecture

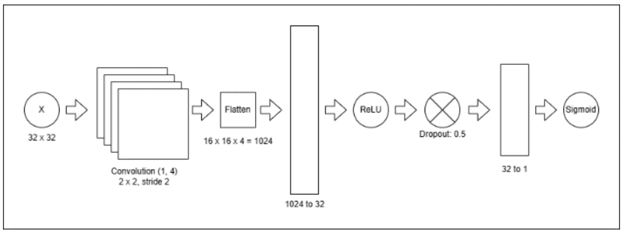


Figure 8 - CNN Architecture

In a sequential process, each image undergoes processing by the quanvolutional layer. The 2×2 filter is applied to the image, and for each filter application, a random circuit is generated to extract features. The resulting feature maps have dimensions of $(14, 14, 4)$ for the downsampled dataset and $(16, 16, 4)$ for the original dataset. To facilitate future use without regenerating the features, the feature maps are saved in an NPY file.

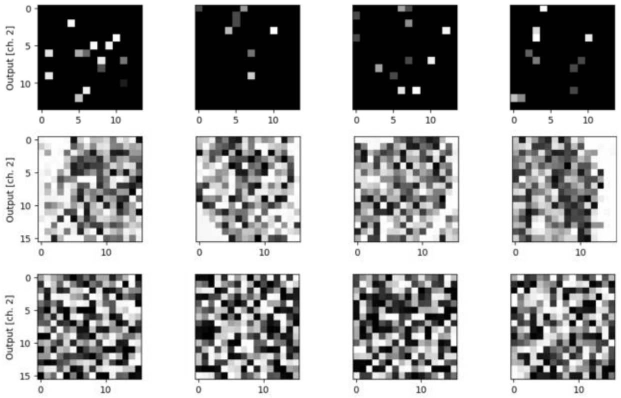


Figure 9 – Quanvolutional Layer Extracted Features: downsampled (top), original (center), MedViT Features (down)

The process of training and evaluating the two architectures on the datasets involves the application of 10 distinct and randomly generated splits. The predetermined proportion for these splits is 80% for training data and 20% for test data. For each split, a K-Fold cross-validation with $K = 5$ is executed to train and evaluate on the training data. The number of epochs is fixed at 30. This procedure yields average training and validation losses, along with average accuracies on the validation partitions for each epoch. Using this data, the optimal number of epochs is determined based on the epoch with the minimum validation loss. Subsequently, a new model is trained on the entire training data using the optimal number of epochs. This finalized model undergoes evaluation on the test dataset. In all cases, a binary cross-entropy loss function is employed with an Adam optimizer. The specific hyperparameters utilized are detailed in the subsequent section. For each train-test split, metrics such as accuracy, precision, recall, F1-measure, and the confusion matrix are computed on the test dataset. These values are aggregated and averaged across all splits. Additionally, the training and validation loss curves, as well as the validation accuracy curves, are averaged over all splits to generate the overall average curves for the entire training process. The entire training and evaluation pipeline is replicated with identical parameters for the second architecture. However, in this case, the features extracted by the quanvolutional layer serve as inputs.

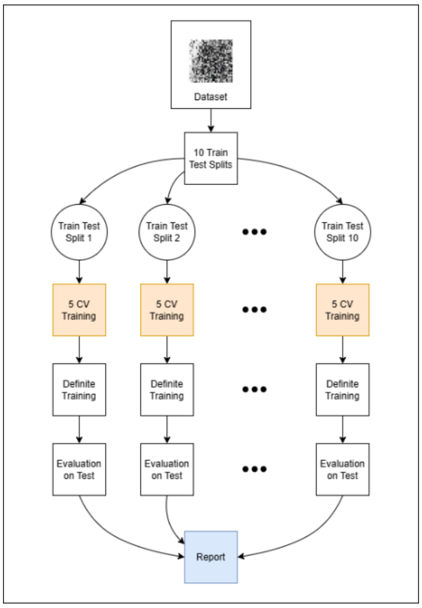


Figure 10 – QNN/CNN Training Pipeline

Several variations of this comprehensive process are introduced. In one experiment, instead of utilizing the original dataset images directly, they undergo preprocessing with MedViT, a visual transformer pre-trained on histological images [16]. Subsequently, these features are fed into both the quanvolutional and convolutional layers. The purpose is to evaluate whether the performance of the quanvolutional model improves by incorporating these features. Another variation incorporates a data augmentation component, effectively doubling the size of the original dataset. This augmentation involves a single iteration of random horizontal flips and rotations within the range of -15 to 15 degrees, resulting in 400 images per class. Once again, the aim is to observe a potential enhancement in the quanvolutional model's performance. Yet another experiment involves the addition of a convolutional layer to each of the models. In the classic convolutional model, a second layer is introduced immediately following the first, receiving 4 channels and outputting 16. It maintains the same filter size (2×2) and a stride of 2. The quanvolutional counterpart of this new layer comprises a second quantum circuit. This circuit mirrors the first but incorporates 16 qubits and concludes with 16 measurements. Consequently, it accommodates 4 filters of size 2×2, producing a measurement for each rotated qubit. Random gates are interposed between measurements, and the encoding utilizes RY gates, as described previously. The ultimate experiment involves the original dataset with one convolutional layer in each architecture, incorporating hyperparameter tuning through grid search. A predefined set of values for specific hyperparameters is examined, with the objective of optimizing the F1-Measure. The parameters utilized for each of these experiments are delineated in the subsequent section.

Another experiment involves incorporating a variational quantum circuit onto a ResNet18 architecture, aiming to implement transfer learning. The quantum circuit comprises three layers: the first initializes the received qubits in a balanced superposition using Hadamard gates, followed by a set of RY gates that perform rotations based on input parameters, akin to the quanvolutional layer in previous experiments. The second layer is variational and trainable, consisting of an entangling sub-layer utilizing CNOT gates and additional rotations with RY. Finally, the third layer measures the rotated qubits in the Z basis. This quantum circuit is integrated into a Dressed Quantum Neural Network, featuring a classical pre-processing linear layer that receives features from the pre-trained model. This layer applies tanh as an activation function, performs scaling, and transmits the resultant vectors as qubits to the variational circuit. A classical post-processing step is then applied to the values obtained from the measurements. The classical optimizer for this variational quantum circuit is Adam, and the entire training process operates within a standard Torch training loop utilizing Binary Cross Entropy as the loss function. In a distinct experiment, MedViT features are utilized as inputs for the Dressed Quantum Neural Network instead of using ResNet18 as the feature extractor.

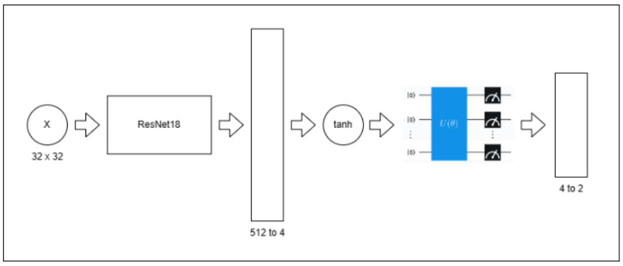


Figure 11 - ResNet18 Transfer Learning to VQC (Dressed Quantum Neural Network)

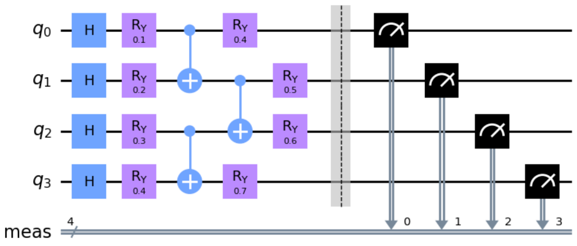


Figure 12 - An example of the VQC used. This one has depth 1 but in reality depth 6 was used.

V. EXPERIMENTS

A total of six experiments were conducted. Experiment 1 utilized the downsampled dataset featuring images of dimensions (28×28), while Experiment 2 employed the original dataset with images resized to (32×32). In Experiment 3, the features extracted from the MedViT pre-trained transformer were employed. These features, initially vectors of size 1024, were reshaped to (32×32) to facilitate the application of convolutions. Experiment 4 involved the original dataset augmented by a single iteration, resulting in a dataset twice the size of the original. Experiment 5 utilized the original dataset but incorporated an additional convolutional layer into each architecture.

Exp.	Dataset	Conv. Layers
1	Downsampled (28 x 28)	1
2	Original (32 x 32)	1
3	MedViT Features (32 x 32)	1
4	Original Augmented (32 x 32)	1
5	Original (32 x 32)	2

In each of these five experiments, the following hyperparameters were configured as stated below:

Hyperparameter	Value
Training Data Proportion	80%
Test Data Proportion	20%
Different Train Test Splits	10
Train Test Splits Seeds	1, 2, …, 10
Folds (K)	5
Loss Function	Binary Cross Entropy
Optimizer	Adam
Cross-Validation (CV) Epochs	30
Definite Training Epochs	CV Minimum Loss Epoch
Cross-Validation Folds Seed	42
Batch Size	32
L_2 Regularization Rate	0.01
Learning Rate	0.001
Dropout Rate	0.5
MLP Hidden Layer Size	32
Convolution 1 Input Channels	1
Conv. 1 Output Channels	4
Q-Circuit Random Layers	1
Q-Circuit Params. Seed	0

Regarding Experiment 5, the parameters for the second convolution are as follows:

Hyperparameter	Value
Conv. 2 Input Channels	4
Conv. 2 Output Channels	16

Another concluding experiment involves applying one convolutional layer to each architecture in the original dataset, incorporating hyperparameter tuning through grid search this time.

Exp.	Dataset	Conv. Layers
6	Original (32 x 32)	1

The hyperparameters slated for tuning and the range of values explored in Experiment 6 are outlined below:

Hyperparameter	Explored Values
L_2 Regularization Rate	0.01, 0.001
Learning Rate	0.001, 0.005
Dropout Rate	0.3, 0.5, 0.7
MLP Hidden Layer Size	32, 64, 128

The final two experiments were related to the transfer learning task. The first one using the ResNet18 pre-trained architecture as a feature extractor for the Dressed Quantum Neural Network and the second one using the MedViT features as inputs for this variational quantum circuit. The specific hyperparameters used for these experiments are presented below:

Hyperparameter	Value
VQC Qubits	4
Learning Rate	0.0004
Batch Size	4
Epochs	8, 30
VQC Depth	6
Learning Rate Reduction	0.1, 10 epochs
Optimizer	Adam
Loss Function	Binary Cross Entropy
Initial VQC Params. Spread	0.01
Training Data Proportion	80%
Test Data Proportion	20%
Random Numbers Seed	42

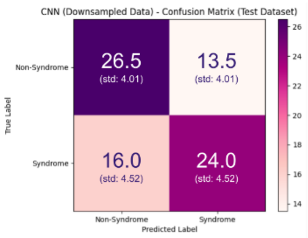
VI. RESULTS

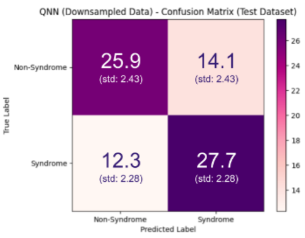
In this section, the results obtained from each of the experiments are presented.

Experiment 1: Downsampled Dataset

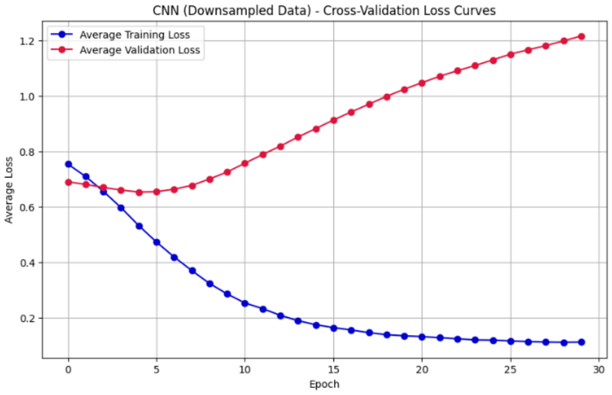
CNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.63125000	0.06027282
Precision	0.64379188	0.06654626
Recall	0.60000000	0.11291590
F1-Measure	0.61526294	0.07671040

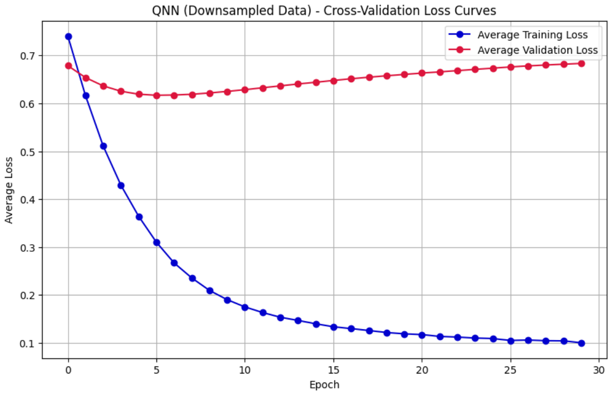
QNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.67000000	0.04115519
Precision	0.66396559	0.04328350
Recall	0.69250000	0.05706356
F1-Measure	0.67682541	0.04178743

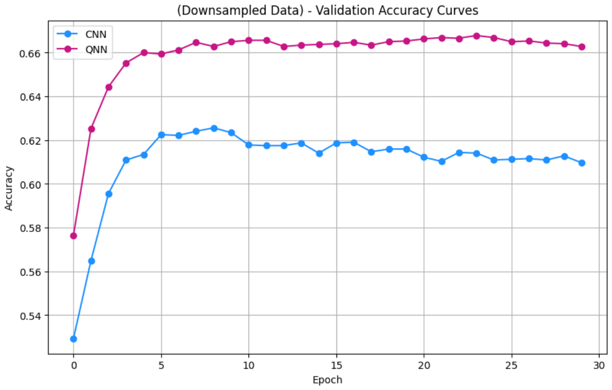


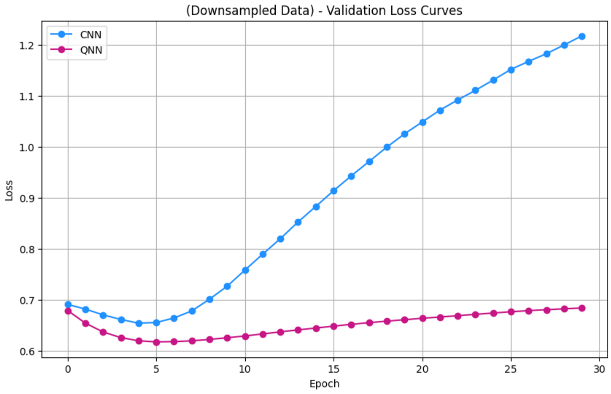


The outcomes from these experiments indicate superior performance of the QNN architecture compared to the CNN architecture in both the test and validation datasets. Overfitting becomes evident after the 5th epoch within the cross-validation training loop. Notably, an intriguing aspect of this experiment is the consistent accuracy observed despite the growing number of epochs, a trend not observed in other subsequent experiments.





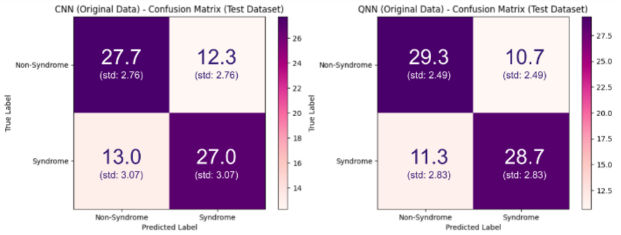




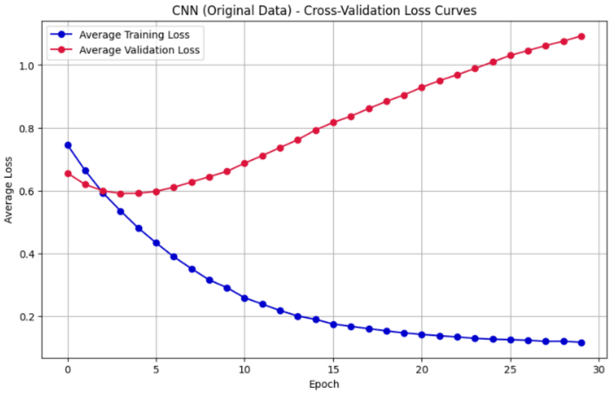
Experiment 2: Original Dataset

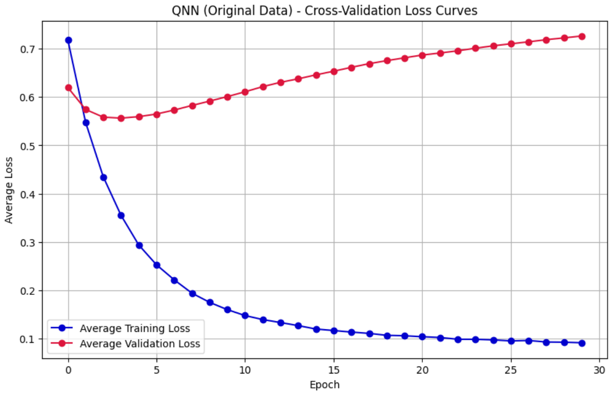
CNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.68375000	0.04403479
Precision	0.68867611	0.04511589
Recall	0.67500000	0.07664855
F1-Measure	0.67943166	0.05205518

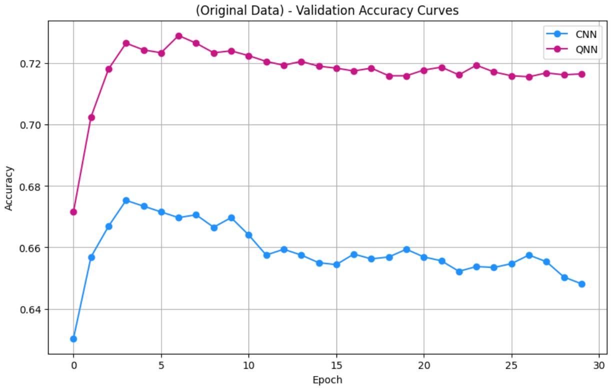
QNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.72500000	0.05031153
Precision	0.72934754	0.05366285
Recall	0.71750000	0.07075486
F1-Measure	0.72203763	0.05587977

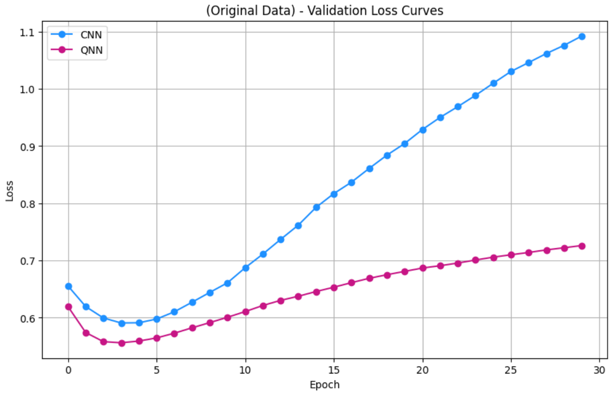


In this subsequent experiment, the models achieve an elevated level of performance in comparison to the preceding one. However, the QNN continues to outperform the CNN. Overfitting becomes apparent after 2 or 3 epochs within the cross-validation training loop. The accuracy experiences a slight decline with the progression of epochs in the cross-validation training loop.







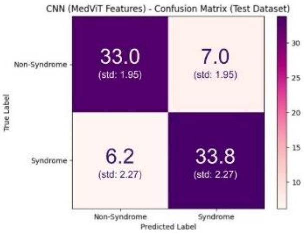


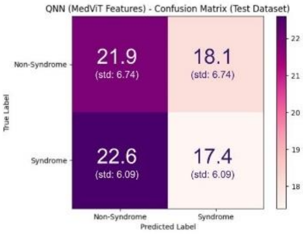
Experiment 3: MedViT Features

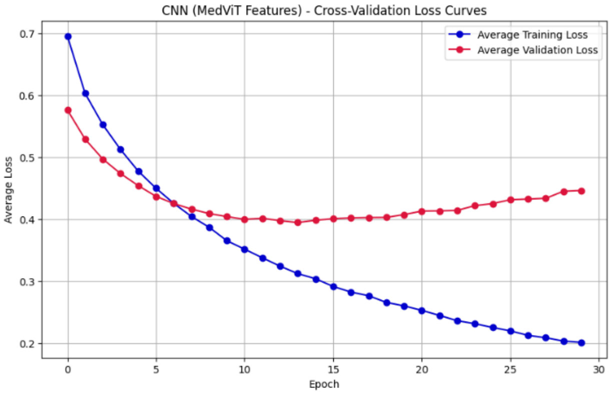
CNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.83500000	0.03000000
Precision	0.83042896	0.03712276
Recall	0.84500000	0.05678908
F1-Measure	0.83610966	0.03178865

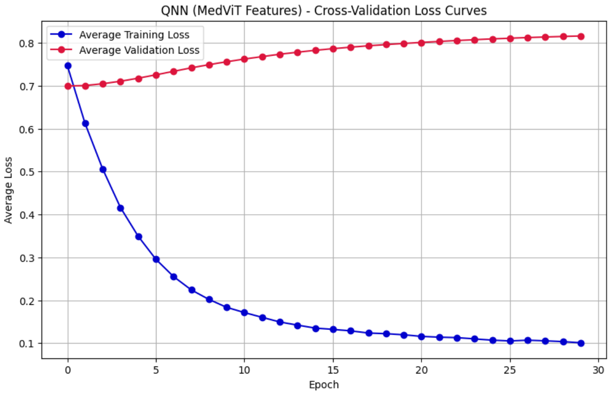
QNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.49125000	0.03012993
Precision	0.49649785	0.05420779
Recall	0.43500000	0.15215124
F1-Measure	0.44762876	0.08414821

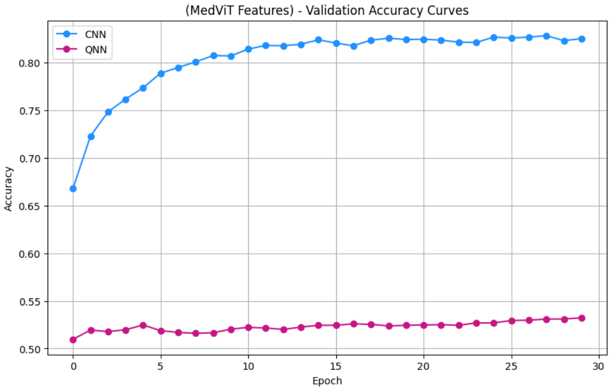
This experiment reveals a markedly improved performance for the CNN, surpassing 80% in all metrics. As anticipated, this improvement is attributed to the features extracted from MedViT, a vision transformer pre-trained on histopathological images. Surprisingly, the QNN experiences a significant decline in performance, falling below 50%, and the reason for its inability to capture MedViT features, as effectively as the classical convolutional layer, remains unclear. In the CNN architecture, overfitting becomes noticeable after 13 epochs. Notably, the validation loss curve for this architecture in this experiment grows at a much slower rate than in other experiments, suggesting that MedViT holds promise for effectively classifying this dataset.

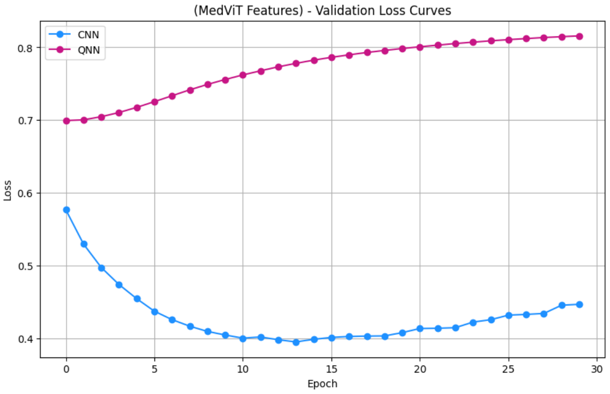












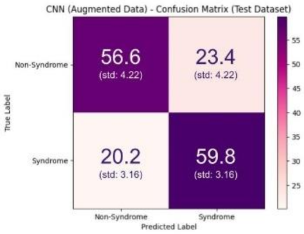
Experiment 4: Augmented Data

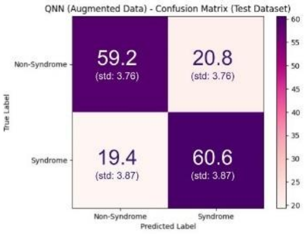
CNN Average Performance on the Test Dataset

Metric	Average	Std. Dev.
Accuracy	0.72750000	0.02549510
Precision	0.72049573	0.03360504
Recall	0.74750000	0.03944933
F1-Measure	0.73274624	0.02445090

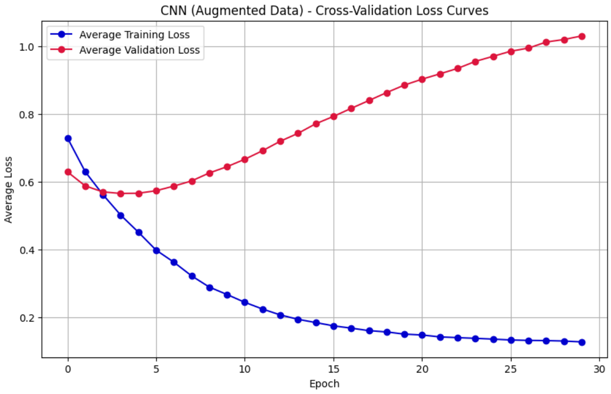
QNN Average Performance on the Test Dataset

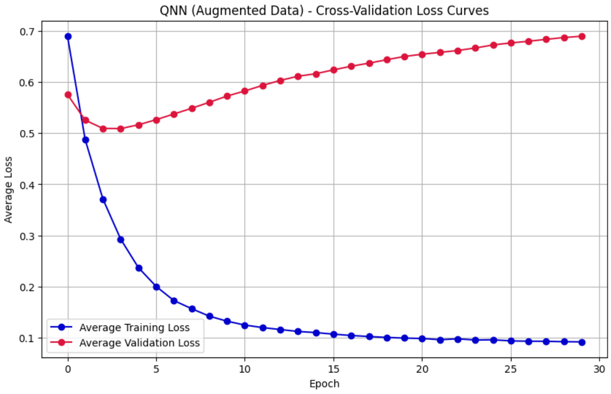
Metric	Average	Std. Dev.
Accuracy	0.74875000	0.03398069
Precision	0.74547040	0.03694686
Recall	0.75750000	0.04847680
F1-Measure	0.75062043	0.03507490

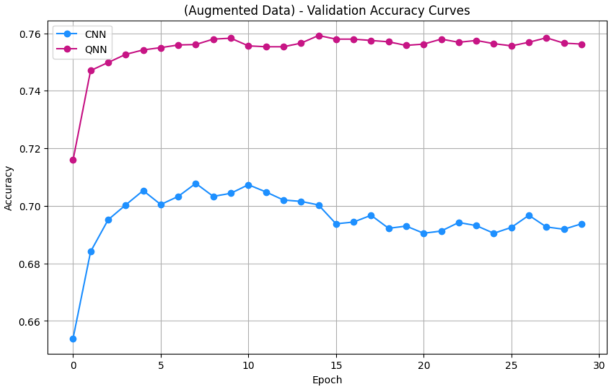


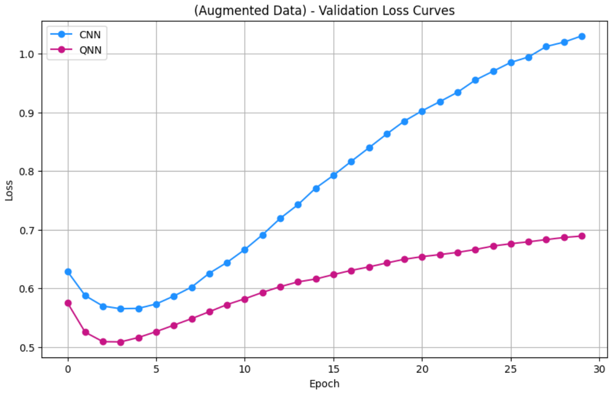


This experiment demonstrates a notable improvement in the classical CNN architecture's performance. Comparing the accuracy of the CNN in Experiment 2, it rose from 0.68375 to 0.74875. The QNN's performance also saw an increase, with the accuracy in Experiment 2 at 0.725 and reaching 0.74875 in this experiment. However, the rate of improvement is less pronounced than that achieved by the CNN. This raises the question of whether there exists a certain number of data augmentation iterations where the CNN starts to outperform the QNN. Additionally, this experiment reveals the accuracy maintaining a relatively stable trend with the increasing number of epochs in the cross-validation training loop.





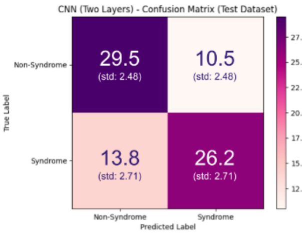


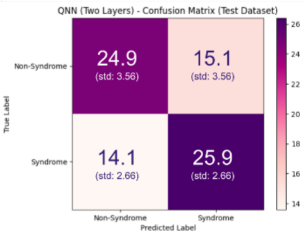


Experiment 5: Two Convolutional Layers:

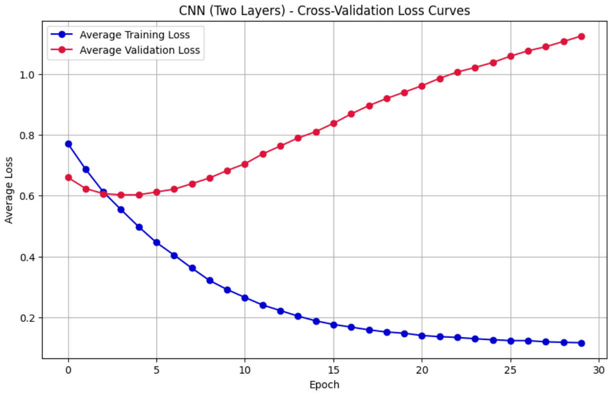
CNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.69625000	0.04147665
Precision	0.71524554	0.04614958
Recall	0.65500000	0.06782330
F1-Measure	0.68199583	0.04917103

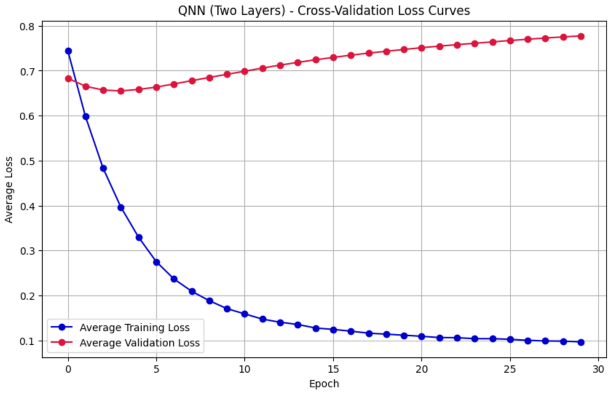
QNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.63500000	0.04286607
Precision	0.63582378	0.05159591
Recall	0.64750000	0.06656763
F1-Measure	0.63882870	0.04160278

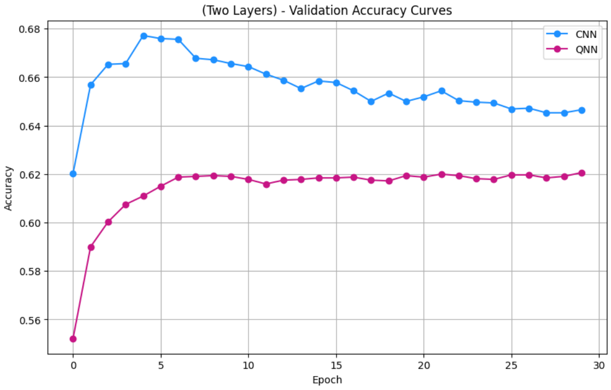


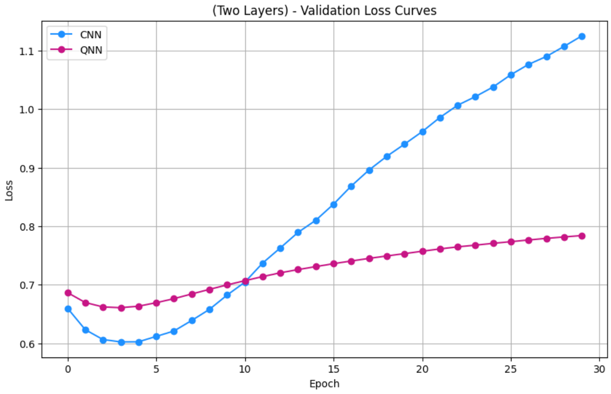


This experiment reveals that the addition of a second QNN layer to the QNN architecture leads to a reduction in performance. The reason for this decline is not immediately evident. It is conceivable that altering the number of input and output channels in this layer may yield different results. Further exploration is necessary when constructing a QNN layer that returns fewer channels than the input channels. In this experiment, it is observed that the validation set's loss increases at a slower rate for the QNN compared to the CNN, indicating that the second classical convolutional layer contributes more to overfitting than the second quanvolutional layer.









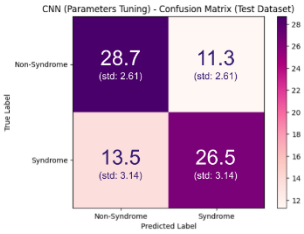
Experiment 6: Hyperparameters Tuning:

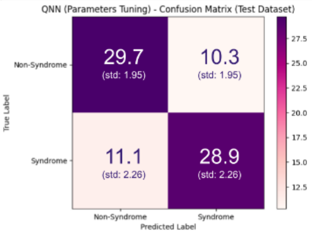
CNN Grid Search Results	
Parameter	**Value**
Learning Rate	0.001
Dropout Rate	0.3
L_2 Regularization Rate	0.01
MLP Hidden Layer Size	128
Best F1-Measure	0.72994769

CNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.69000000	0.03699662
Precision	0.70399016	0.04454081
Recall	0.66250000	0.07846177
F1-Measure	0.67933588	0.04724284

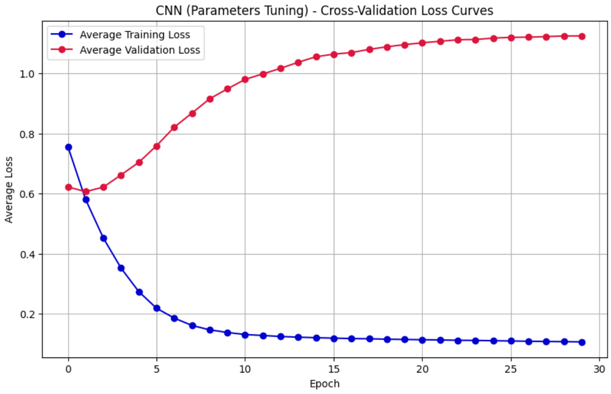
QNN Grid Search Results	
Parameter	**Value**
Learning Rate	0.001
Dropout Rate	0.5
L_2 Regularization Rate	0.01
MLP Hidden Layer Size	32
Best F1-Measure	0.75269634

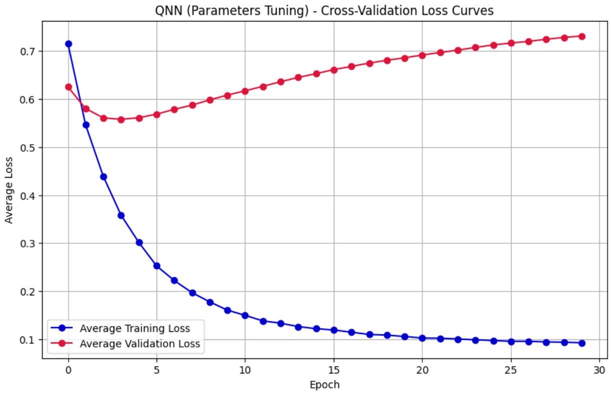
QNN Average Performance on the Test Dataset		
Metric	**Average**	**Std. Dev.**
Accuracy	0.73250000	0.03674235
Precision	0.73832343	0.04004196
Recall	0.72250000	0.05640257
F1-Measure	0.72920578	0.03978096

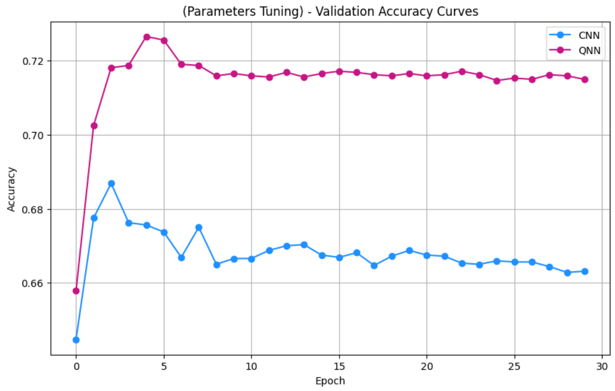


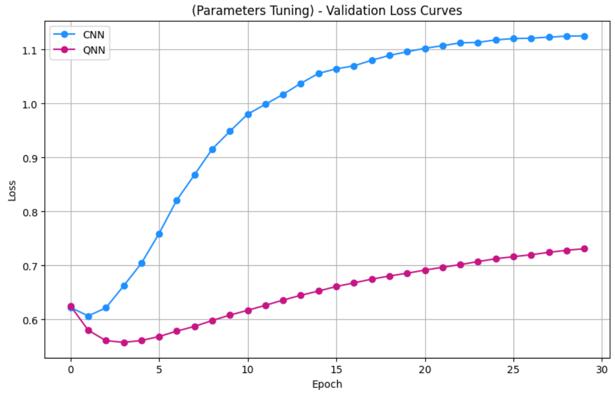


This experiment shows that, even when using grid search, the QNN performs better than the CNN, achieving the best F1-Measure. However, when using the identified parameters to train the models, the resulting metrics are lower. This difference may be because, during the exploration, five different train-test splits are used, while during actual training, ten different splits are used. It's important to note that the grid search exploration doesn't seem very effective, as the results are similar to those in Experiment 2. This could be because the search space used for exploration wasn't broad enough to find optimal parameters, or it might be that the predefined architectures can't achieve better results on this dataset. Performing hyperparameter tuning with the augmented dataset could be beneficial. For the classical model, trying data augmentation before extracting MedViT features and then conducting grid search before training might lead to even better results and is worth exploring.



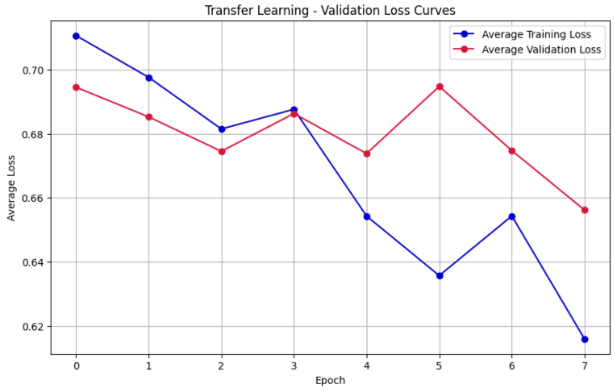


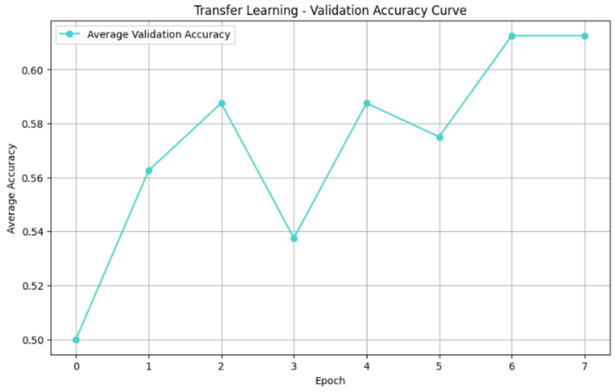




Experiment 7: ResNet18 Transfer Learning to VQC:

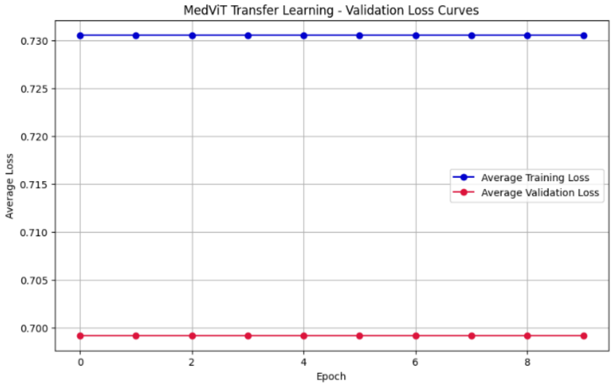
In this experiment, the Variational Quantum Circuit (VQC) achieves an accuracy of 0.67 after training for 8 epochs. However, as the number of epochs is increased, the accuracy begins to fluctuate and, in some cases, even decreases.

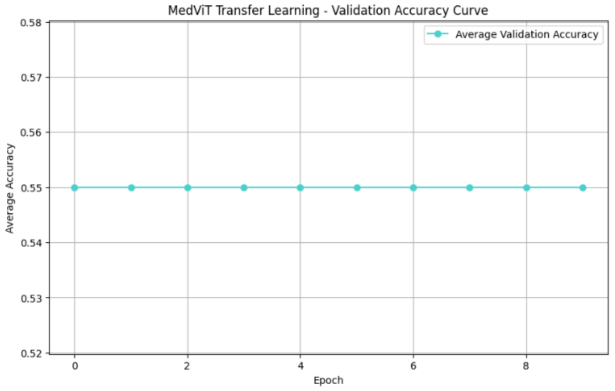




Experiment 8: MedViT Transfer Learning to VQC:

In this experiment, it was noticed that the Variational Quantum Circuit (VQC) did not learn from the features extracted by the MedViT architecture. The accuracy and loss remained constant throughout the entire training process, even when the number of epochs was increased to over 10. The reason why the Dressed Quantum Neural Network (VQC) struggles to learn from these features is not evident, and a similar observation was made in Experiment 3, where the QNN exhibited significantly lower performance when using these features. This stands in contrast to the classical CNN, raising an intriguing question for further exploration in future studies.





VII. CONCLUSIONS

The findings from the conducted experiments reveal notable insights into the performance of Quantum Neural Network (QNN) architectures in comparison to Classical Convolutional Neural Network (CNN) architectures across various scenarios. In the initial set of experiments, the QNN consistently demonstrated superior performance over the CNN in both test and validation datasets, showcasing a remarkable and sustained accuracy, particularly noteworthy due to its stability with the increasing number of epochs. This trend, however, was not consistently observed in subsequent experiments. In subsequent experiments, the models achieved higher performance compared to preceding ones, yet the QNN continued to outperform the CNN. Overfitting patterns became evident in these experiments, manifesting after a certain number of epochs within the cross-validation training loop. A significant observation arose in an experiment utilizing features extracted from MedViT. The CNN exhibited a substantial performance boost, surpassing 80% in all metrics, while the QNN experienced a notable decline, falling below 50%. This raises questions about the QNN's capability to effectively capture features from MedViT, a phenomenon not observed in the classical convolutional layer. Further exploration introduced the notion of data augmentation iterations impacting the CNN's outperformance over the QNN. Additionally, the addition of a second QNN layer led to a reduction in performance, with considerations for altering input and output channels in this layer for improved results. Grid search exploration exhibited the QNN's superior F1-Measure, although model training with the identified parameters yielded lower metrics, prompting considerations about the effectiveness of the exploration. The introduction of a Variational Quantum Circuit (VQC) revealed fluctuating accuracy during training, highlighting potential challenges in learning from MedViT features. This phenomenon echoed a similar observation in Experiment 3 involving the QNN. Contrastingly, the classical CNN maintained stability in performance. Future studies are suggested to explore variations in the quanvolutional layer's random layers, experiment with different pre-set quantum circuits, and investigate the impact of changing random parameters for the quantum circuit in each execution. These avenues of exploration hold promise for further understanding and optimizing the performance of quantum architectures in image classification tasks.

REFERENCES

[1] H, Eratlay K, Kausum E y Ebersole JL. Periodontal status and serum antibody responses to oral microorganisms in Sjögren´s syndrome. J Periodontol, 1998;69: 571-577.

[2] Greenspan D. Xerostomia: Diagnosis and Management. Oncology, 1996; 10 (3): 7-11.

[3] Vega Castro, Kelly Patricia, & Pertuz Charris, Neil Smith. (2020). Diagnóstico diferencial dentro del espectro del síndrome de Sjögren. Revista Colombiana de Reumatología, 27(Suppl. 2), 67-74. Epub September 10, 2021.https://doi.org/10.1016/j.rcreu.2020.06.002

[4] Shmatko, A., Ghaffari Laleh, N., Gerstung, M. et al. Artificial intelligence in histopathology: enhancing cancer research and clinical oncology. Nat Cancer 3, 1026–1038 (2022). https://doi.org/10.1038/s43018-022-00436-4

[5] B. Acs, M. Rantalainen, J. Hartman, Artificial intelligence as the next step towards precision pathology, J. Int. Med., 288 (2020), 62–81. https://doi.org/10.1111/joim.13030 doi: 10.1111/joim.13030

[6] clínica mayo (articulo web) (agosto 02-2022) https://www.mayoclinic.org/es/diseases-conditions/sjogrens-syndrome/symptoms-causes/syc-20353216

[7] Rivera, H, Valero, L, Escalona, L, Roja-Sánchez, F, & Ríos, MP. (2009). Manejo Multidisciplinario del paciente diagnosticado con el Síndrome de Sjögren. Acta Odontológica Venezolana, 47(3), 122-130. Recuperado en 07 de septiembre de 2023, de http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0001-63652009000300017&lng=es&tlng=es.

[8] Parra-Medina R, Polo J, Rojas A. Interpretación de la biopsia de glándula salival menor en el síndrome de Sjögren. Correlación clínico-patológica. Rev Colomb Reumatol. 2020 Oct; 27 Suppl: 82-89 doi: 10.1016/j.rcreu.2020.05.008.

[9] C.-M. Lo and P.-H. Hung, “Computer-aided diagnosis of ischemic stroke using multi-dimensional image features in carotid color doppler,” Computers in Biology and Medicine, vol. 147, p. 105779, 2022.

[10] S. Igarashi, Y. Sasaki, T. Mikami, H. Sakuraba, and S. Fukuda, “Anatomical classification of upper gastrointestinal organs under various image capture conditions using alexnet,” Computers in Biology and Medicine, vol. 124, p. 103950, 2020.

[11] R. Togo, H. Watanabe, T. Ogawa, and M. Haseyama, “Deep con volutional neural network-based anomaly detection for organ clas sification in gastric x-ray examination,” Computers in biology and medicine, vol. 123, p. 103903, 2020

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020.

[13] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” arXiv preprint arXiv:2102.12122, 2021

[14] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu et al., “A survey on vision transformer,” IEEE transactions on pattern analysis and machine intelligence, 2022

[15] Sirawit Butmaratthaya, Niti Buesamae, Unchalisa Taetragool; MNIST quantum classification models implementation and benchmarking. AIP Conf. Proc. 7 November 2023; 2906 (1): 070002. https://doi.org/10.1063/5.0178776

[16] Omid Nejati Manzari, Hamid Ahmadabadi, Hossein Kashiani, Shahriar B. Shokouhi, Ahmad Ayatollahi, MedViT: A robust vision transformer for generalized medical image classification, Computers in Biology and Medicine, Volume 157, 2023, 106791, ISSN 0010-4825, https://doi.org/10.1016/j.compbiomed.2023.106791