MULTI CLASS CLASSIFICATION USING QUANTUM INSPIRED NEURAL NETWORKS FOR CEREBELLAR ATAXIA

ABSTRACT

In this dissertation, a quantum-inspired neural network model, QCNN, to enhance multi-class classification of cerebellar ataxia subtypes. This proposed model consists of two main phases; pre-processing and classification based on the QCNN phase. The corresponding learning procedure is implemented via TensorFlowQuantum as a hybrid quantum-classical (variational) model, where quantum output results are fed to softmax cost function with subsequent minimization of it via optimization of parameters of quantum circuit .The datasets were taken from various authentic sources such as Kaggle and OpenNeuro. Three different subtypes were used for this model. Our results demonstrate comparable accuracy of our solution with classical convolutional neural networks with comparable numbers of trainable parameters.

Keywords: QCNN, TensorFlowQuantum, Multiclass Classification, Softmax

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Introduction

Cerebellar ataxia, a heterogeneous group of neurodegenerative disorders, is characterized by the progressive impairment of motor coordination, balance, and gait due to the degeneration of the cerebellum and its associated pathways.Despite advancements in diagnostic methods, accurately categorizing distinct subtypes of cerebellar ataxia remains challenging due to overlapping clinical manifestations and intricate underlying pathophysiology.Multi class classification, a machine learning task that involves categorizing data into multiple classes, offers a promising avenue for distinguishing between different subtypes of cerebellar ataxia.Quantum-inspired neural networks combine elements of both quantum computing and neural networks, enabling them to capture complex relationships within medical data.

Traditional diagnostic approaches often lack precision in subtype identification. This study aims to address this diagnostic gap by investigating the potential of quantum-inspired neural networks as an innovative solution for improving multi class classification of cerebellar ataxia subtypes. By harnessing the unique capabilities of quantum-inspired techniques, the research seeks to enhance the accuracy of subtype identification, leading to advancements in early diagnosis and personalized treatment strategies for cerebellar ataxia patients.

CHAPTER 2

LITERATURE SURVEY

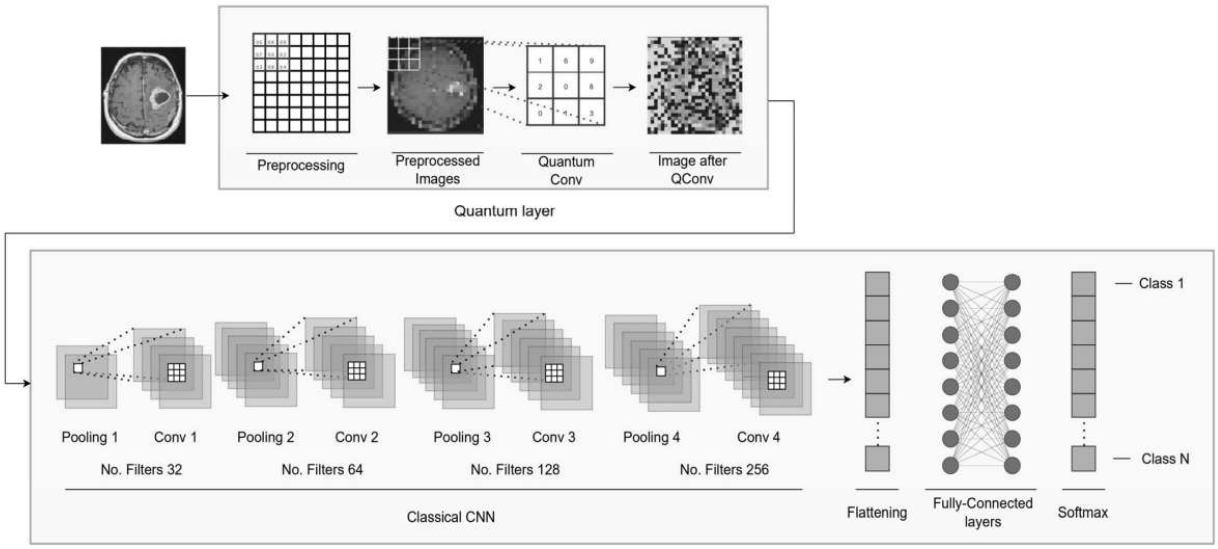
To categorize different types of neuro-degenrative disorders related to spino-cerebellar ataxia several approaches have been used. Firstly the traditional methods have been used which have given good results in a first such as classical CNN's.In a study by **Zhen Yang** et.al[1] a method was propsed for a learning framework using MR image data for discriminating a set of cerebellar ataxia types and predicting a disease related functional score. Another study led by **Oldřich Vyšata** et al.[2] suggested that using reduction methods and classification methods as well, huge amounts of MRI data can led to dimensionally reduced data with better and rich features.Similarly, **Srinivasa Rao Swarna** et al.[6] used machine learning and the quantum computing the proposed system is working on implementing the speech signal-based implementation on the Parkinson's disease prediction.A paper procured by **Hsu SY, Yeh LR** *et al.[9]* compared five convolutional neural networks, to enhance the classification of Parkinson's disease stages using 99mTc-TRODAT-1 SPECT images, achieving the best accuracy with VGG in the six-stage model using color images.In another study by **Maolin Wang** et al.[15] proposed design that performs inference operations on a stream of individual images as they are produced and has a deeply pipelined hardware design that allows all layers of a quantized convolutional neural network (QCNN) to compute concurrently with partial image inputs.The researchers explore how quantum computing concepts, such as superposition and entanglement, can be harnessed to improve computational efficiency, solve certain problems faster, and address challenges faced by classical deep learning methods. The research often delves into comparing the performance of quantum and classical models, as well as exploring the theoretical foundations and practical implications of incorporating quantum principles into deep learning architectures.

CHAPTER 3

PROPOSED METHODOLOGY

3.1. Proposed Methodology

The two primary phases of the proposed image classification model are the pre-processing phase and the classification based on the QCNN phase. The overall design of the proposed image classification model is depicted in the figure below. From this figure, it can be seen that the original images are first normalized, after which the normalized images are fed into the quantum convolutional layer, after which a series of classical convolutional layers are applied, followed by a fully connected layer to obtain the final class. The following sections will go over each part's full description.



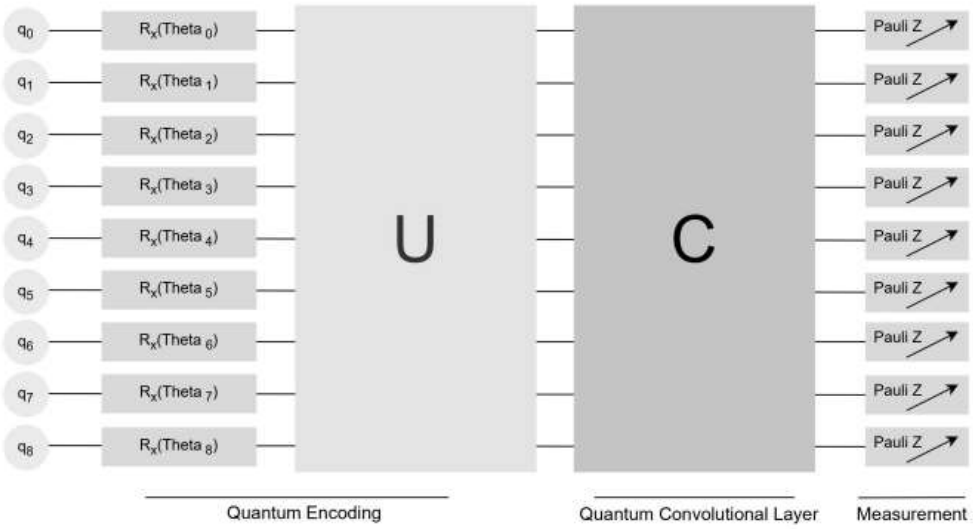
MODEL ARCHITECTURE

3.1.1. Pre-Processing Phase

Preparing and transforming the raw dataset into a more acceptable format for subsequent analysis and modeling is the primary goal of the pre-processing phase. The original images in this study are downsized to 28x28, and after that, data normalization is applied to all of the images in the dataset. Data normalization is crucial to improve the effectiveness and interpretability of machine learning algorithms. Normalization is a method that rescales the input features to a uniform range, usually between 0 and 1. This standardized range aids in reducing the influence of data distribution and scale differences, which could otherwise result in biased and less-than-ideal model performance. Each pixel in the image is normalized in this study by multiplying it by the highest number, 255. Each data point in the image is effectively scaled down to a value between 0 and 1 when this procedure is done to the entire dataset's image. The model is made to be less sensitive to the input characteristics' absolute values and more focused on the underlying patterns and relationships in the data by doing this normalization.

3.1.2. Classification Based on QCNN Phase

This phase feeds the processed image into the quantum convolutional neural network (QCNN). The quantum and classical components of the proposed QCNN make up its two main components. The proposed creates a powerful and effective classification system by combining the advantages of both quantum and conventional methodologies. Next, a detailed description of each part is presented. The quantum part transforms the input data using the capabilities of quantum computing as a preprocessing layer. Quantum encoding, convolution, and measurement comprise its three main components. Quantum Convolutional Neural Networks (QCNN) behave in a similar manner to CCNNs. First, we encode our pixelated image into a quantum circuit using a given feature map, such Qiskit’s ZFeatureMap or ZZFeatureMap or others available in the circuit library.

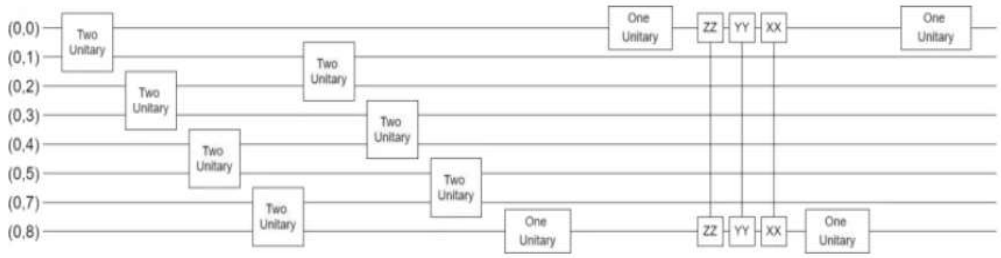


Quantum part of proposed CNN

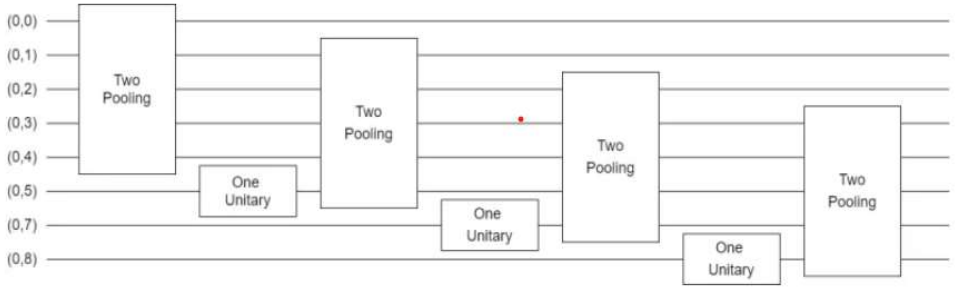
The input data is first embedded into a quantum state during the quantum encoding stage. Nine qubits total are used in this study.

$$R_X(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

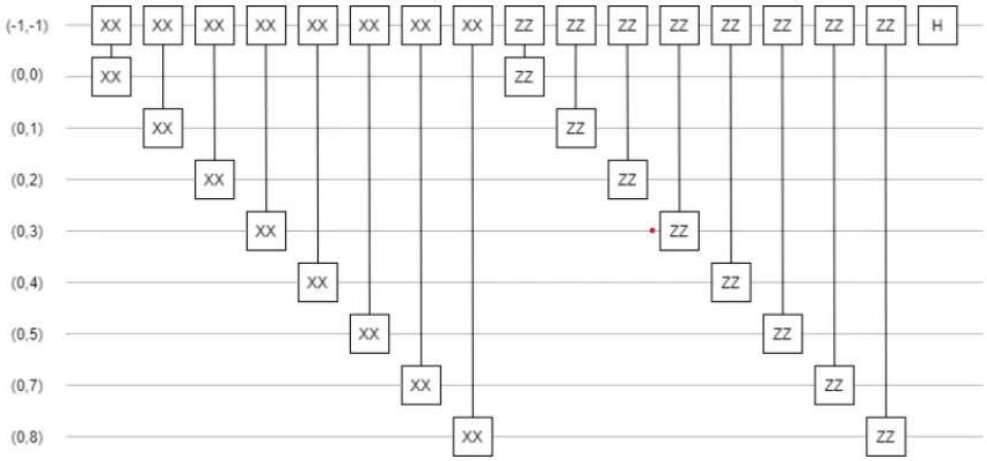
3.1.3. Quantum Convolution Neural Networks



Quantum Convolution Layer



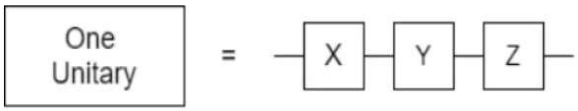
Quantum Pooling Layer



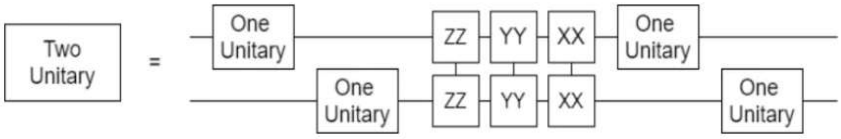
Quantum Neural Network Layer

3.1.4. Quantum Encoding

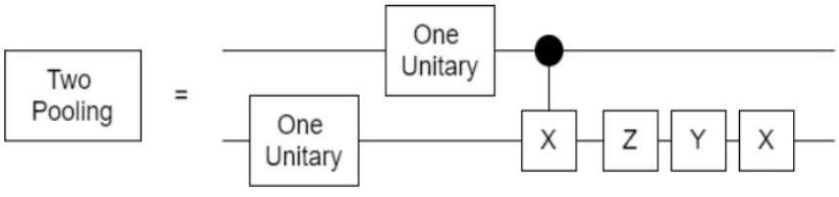
Quantum Convolutional Neural Networks (QCNN) behave in a similar manner to CCNNs. First, we encode our pixelated image into a quantum circuit using a given feature map, such Qiskit's ZFeatureMap or ZZFeatureMap or others available in the circuit library.



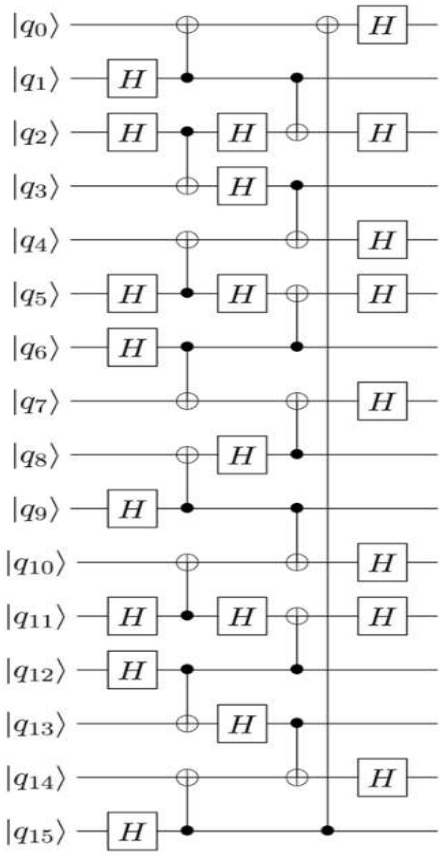
One Cubit Unitary Circuit



Two-qubit Unitary Circuit



Two-qubit Pooling Circuit

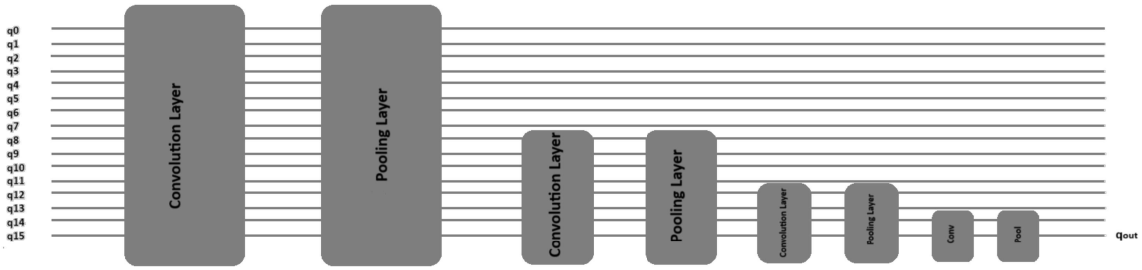


Quantum Circuit

After encoding our image, we apply alternating convolutional and pooling layers, as defined in the next section. By applying these alternating layers, we reduce the dimensionality of our circuit until we are left with one qubit. We can then classify our input image by measuring the output of this one remaining qubit.

The Quantum Convolutional Layer will consist of a series of two qubit unitary operators, which recognize and determine relationships between the qubits in our circuit. This unitary gates are defined below in the next section.For the Quantum Pooling Layer, we cannot do the same as is done classically to reduce the dimension, i.e. the number of qubits in our circuit. Instead, we reduce the number of qubits by performing operations upon each until a specific point and then disregard certain qubits in a specific layer. It is these layers where we stop performing operations on certain qubits that we call our 'pooling layer'. Details of the pooling layer is discussed further in the next section. In the QCNN, each layer

contains parametrized circuits, meaning we alter our output result by adjusting the parameters of each layer. When training our QCNN, it is these parameters that are adjusted to reduce the loss function of our QCNN.
An example of 16 bit qubit QCNN can be seen below.



CNN NETWORK WITH SINGLE QUBIT OUTPUT

3.1.5. Convolutional Layer

The next step in this tutorial is to define the Convolutional Layers of our QCNN. These layers are then applied to the qubits after the data has been encoded through use of the feature map.To do so we first need to determine a parametrized unitary gate, which will be used to create our convolutional and pooling layers.A small region of the input image, in our example a 28x 28 square, is embedded into a quantum circuit. In this demo, this is achieved with parametrized rotations applied to the qubits initialized in the ground state. A quantum computation, associated to a unitary U, is performed on the system. The unitary could be generated by a

variational quantum circuit or, more simply, by a random circuit. This is achieved with parametrized rotations applied to the qubits initialized in the ground state. A quantum computation, associated to a unitary U, is performed on the system. The unitary could be generated by a variational quantum circuit or, more simply, by a random circuit. The quantum system is finally measured, obtaining a list of classical expectation values. The measurement results could also be classically post-processed but, for simplicity, we directly use the raw expectation values. Analogously to a classical convolution layer, each expectation value is mapped to a different channel of a single output pixel. Iterating the same procedure over different regions, one can scan the full input image, producing an output object which will be structured as a multi-channel image. The quantum convolution can be followed by further quantum layers or by classical layers. The main difference with respect to a classical convolution is that a quantum circuit can generate highly complex kernels whose computation could be, at least in principle, classically intractable.

3.1.6. Classical Part

We create a function for our QCNN, which will contain three sets of alternating convolutional and pooling layers, which can be seen in the schematic below. Through the use of the pooling layers, we thus reduce the dimensionality of our QCNN from sixteen qubits to one.After the application of the quantum convolution layer we feed the resulting features into a classical neural network that will be trained to classify the three different subtypes of the ataxia dataset.We use a very simple model: just a fully connected layer with 10 output nodes with a final softmax activation function.The model is compiled with a stochiastic-gradient-descent optimizer, and a cross entropy loss function.

CHAPTER 4

Dataset Description

4.1. About Dataset

The datasets have been compiled from various sources such as Kaggle and OpenNeuro. The image that have been used in this model are of types such as SCA-1,SCA-2 and SCA-3. These images were in the NIFTI format from where they have been transformed from 3d to 2d images. Majority of the images were in the highest resolution. Dataset comprised of around 2.1GB unzipped. The dataset was divided into test, train and validation sets eventually.The datset used is was not normalised. The dataset needed to be cropped in order to reduce the dimensions of the images.

CHAPTER 5

EXPERIMENTAL RESULTS

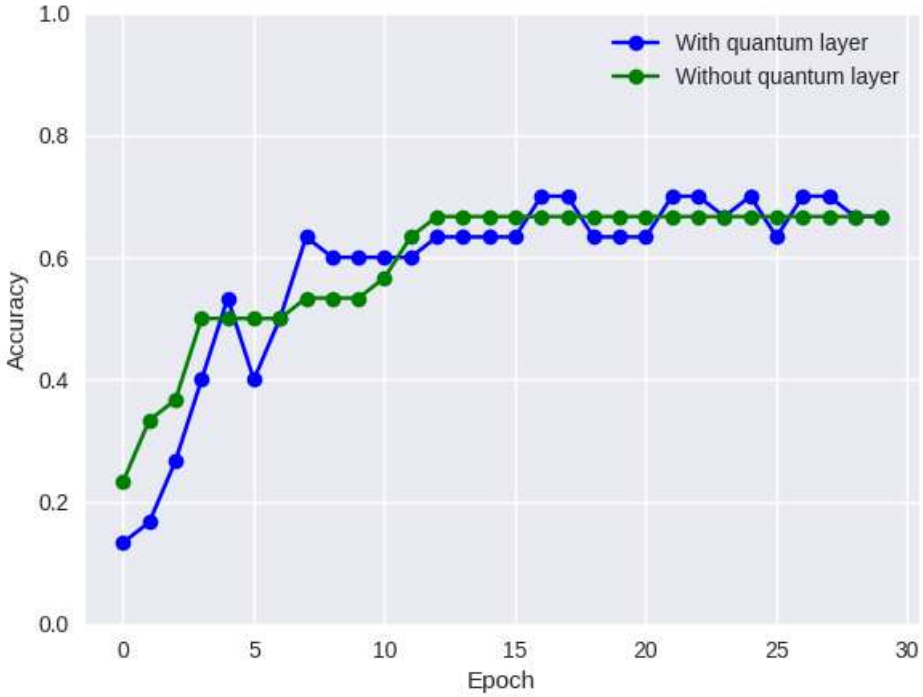
5.1.1 Training Results.

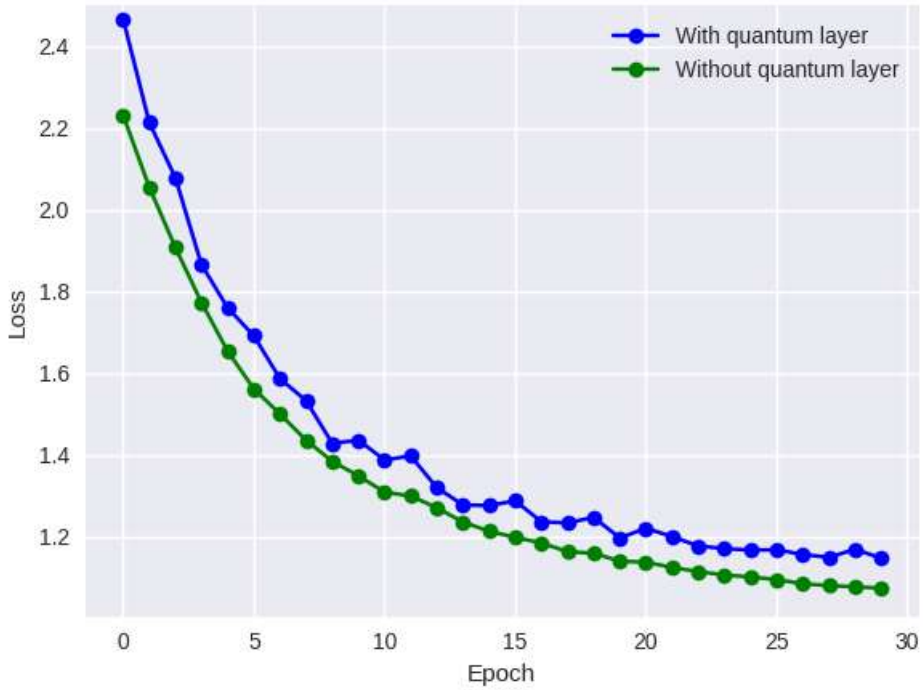
```
Epoch 1/30
13/13 - 0s - loss: 2.8401 - accuracy: 0.1400 -
val_loss: 2.4667 - val_accuracy: 0.1333 - 411ms/epoch
- 32ms/step
Epoch 2/30
13/13 - 0s - loss: 2.1788 - accuracy: 0.1600 -
val_loss: 2.2135 - val_accuracy: 0.1667 - 29ms/epoch
- 2ms/step
Epoch 3/30
13/13 - 0s - loss: 1.8027 - accuracy: 0.4000 -
val_loss: 2.0782 - val_accuracy: 0.2667 - 29ms/epoch
- 2ms/step
Epoch 4/30
13/13 - 0s - loss: 1.4389 - accuracy: 0.6800 -
val_loss: 1.8654 - val_accuracy: 0.4000 - 29ms/epoch
- 2ms/step
Epoch 5/30
13/13 - 0s - loss: 1.2171 - accuracy: 0.7600 -
val_loss: 1.7604 - val_accuracy: 0.5333 - 29ms/epoch
- 2ms/step
Epoch 6/30
13/13 - 0s - loss: 1.0139 - accuracy: 0.8800 -
val_loss: 1.6917 - val_accuracy: 0.4000 - 30ms/epoch
- 2ms/step
Epoch 7/30
13/13 - 0s - loss: 0.8133 - accuracy: 0.8800 -
val_loss: 1.5871 - val_accuracy: 0.5000 - 29ms/epoch
- 2ms/step
Epoch 8/30
13/13 - 0s - loss: 0.6749 - accuracy: 0.9400 -
val_loss: 1.5332 - val_accuracy: 0.6333 - 28ms/epoch
- 2ms/step
Epoch 9/30
13/13 - 0s - loss: 0.5730 - accuracy: 0.9800 -
val_loss: 1.4287 - val_accuracy: 0.6000 - 29ms/epoch
- 2ms/step
Epoch 10/30
13/13 - 0s - loss: 0.4820 - accuracy: 1.0000 -
val_loss: 1.4367 - val_accuracy: 0.6000 - 29ms/epoch
- 2ms/step
Epoch 11/30
```

```
13/13 - 0s - loss: 0.4361 - accuracy: 1.0000 -
val_loss: 1.3882 - val_accuracy: 0.6000 - 29ms/epoch
- 2ms/step
Epoch 12/30
13/13 - 0s - loss: 0.3877 - accuracy: 0.9800 -
val_loss: 1.3980 - val_accuracy: 0.6000 - 29ms/epoch
- 2ms/step
Epoch 13/30
13/13 - 0s - loss: 0.3372 - accuracy: 1.0000 -
val_loss: 1.3206 - val_accuracy: 0.6333 - 30ms/epoch
- 2ms/step
Epoch 14/30
13/13 - 0s - loss: 0.3093 - accuracy: 0.9800 -
val_loss: 1.2783 - val_accuracy: 0.6333 - 29ms/epoch
- 2ms/step
Epoch 15/30
13/13 - 0s - loss: 0.2636 - accuracy: 1.0000 -
val_loss: 1.2766 - val_accuracy: 0.6333 - 29ms/epoch
- 2ms/step
Epoch 16/30
13/13 - 0s - loss: 0.2346 - accuracy: 1.0000 -
val_loss: 1.2884 - val_accuracy: 0.6333 - 29ms/epoch
- 2ms/step
Epoch 17/30
13/13 - 0s - loss: 0.2189 - accuracy: 1.0000 -
val_loss: 1.2370 - val_accuracy: 0.7000 - 29ms/epoch
- 2ms/step
Epoch 18/30
13/13 - 0s - loss: 0.1959 - accuracy: 1.0000 -
val_loss: 1.2342 - val_accuracy: 0.7000 - 29ms/epoch
- 2ms/step
Epoch 19/30
13/13 - 0s - loss: 0.1768 - accuracy: 1.0000 -
val_loss: 1.2477 - val_accuracy: 0.6333 - 29ms/epoch
- 2ms/step
Epoch 20/30
13/13 - 0s - loss: 0.1659 - accuracy: 1.0000 -
val_loss: 1.1975 - val_accuracy: 0.6333 - 29ms/epoch
- 2ms/step
Epoch 21/30
13/13 - 0s - loss: 0.1511 - accuracy: 1.0000 -
```

```
val_loss: 1.2211 - val_accuracy: 0.6333 - 30ms/epoch
- 2ms/step
Epoch 22/30
13/13 - 0s - loss: 0.1379 - accuracy: 1.0000 -
val_loss: 1.2009 - val_accuracy: 0.7000 - 30ms/epoch
- 2ms/step
Epoch 23/30
13/13 - 0s - loss: 0.1304 - accuracy: 1.0000 -
val_loss: 1.1778 - val_accuracy: 0.7000 - 31ms/epoch
- 2ms/step
Epoch 24/30
13/13 - 0s - loss: 0.1216 - accuracy: 1.0000 -
val_loss: 1.1714 - val_accuracy: 0.6667 - 30ms/epoch
- 2ms/step
Epoch 25/30
13/13 - 0s - loss: 0.1164 - accuracy: 1.0000 -
val_loss: 1.1679 - val_accuracy: 0.7000 - 30ms/epoch
- 2ms/step
Epoch 26/30
13/13 - 0s - loss: 0.1054 - accuracy: 1.0000 -
val_loss: 1.1685 - val_accuracy: 0.6333 - 30ms/epoch
- 2ms/step
Epoch 27/30
13/13 - 0s - loss: 0.1002 - accuracy: 1.0000 -
val_loss: 1.1569 - val_accuracy: 0.7000 - 29ms/epoch
- 2ms/step
Epoch 28/30
13/13 - 0s - loss: 0.0977 - accuracy: 1.0000 -
val_loss: 1.1501 - val_accuracy: 0.7000 - 29ms/epoch
- 2ms/step
Epoch 29/30
13/13 - 0s - loss: 0.0899 - accuracy: 1.0000 -
val_loss: 1.1696 - val_accuracy: 0.6667 - 30ms/epoch
- 2ms/step
Epoch 30/30
13/13 - 0s - loss: 0.0836 - accuracy: 1.0000 -
val_loss: 1.1481 - val_accuracy: 0.6667 - 31ms/epoch
- 2ms/step
```

5.1.2 QNN Model performance, in terms of test set accuracy and training- log-loss, compared with CNN model.





epoch	accuracy	loss	val_accuracy	val_loss
1	0.1400	2.8401	0.1333	2.4667
2	0.1600	2.1788	0.1667	2.2135
15	1.0000	0.2636	0.6333	1.2766
29	1.0000	0.0899	0.6667	1.1696
30	1.0000	0.0836	0.6667	1.1481

Loss and accuracy Table

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This study proposed a quantum convolutional neural network architecture. The proposed QCNN is applied to the image classification task. CSA-1,CSA2,CSA-3 Brain MRI Images benchmark datasets are adopted. The experimental results revealed that exploiting quantum principles such as superposition and entanglement in the classical CNN can positively boost its performance. The results showed that increasing the kernel size of the quantum layer can significantly boost the performance of the proposed QCNN. Moreover, the results revealed that the proposed QCNN obtained the best results compared to several well-known landmark models. The overall proposed image classification model achieves an accuracy of 66.7%.The main challenge in implementing the proposed QCNNs to large-scale image classification tasks is the restricted capacity of quantum hardware, which limits the size of input images and the number of layers employed in the network. As a result, in the future, the proposed QCNN may be implemented using simple quantum circuits.

6.2 Future Work

Future work could involve refining quantum-inspired architectures, optimizing feature encoding, exploring hybrid quantum-classical models, developing quantum-friendly optimization strategies, improving interpretability, conducting rigorous validation studies, staying updated on quantum hardware advancements, and addressing ethical considerations like data privacy and bias in clinical deployments.

References

[1] *Yang Z, Zhong S, Carass A, Ying SH, Prince JL. Deep Learning for Cerebellar Ataxia Classification and Functional Score Regression. Mach Learn Med Imaging. 2014;8679:68-76. doi: 10.1007/978-3-319-10581-9_9. PMID: 25553339; PMCID: PMC4278360.*

[2] *Vyšata O, Ťupa O, Procházka A, Doležal R, Cejnar P, Bhorkar AM, Dostál O, Vališ M. Classification of Ataxic Gait. Sensors (Basel). 2021 Aug 19;21(16):5576. doi: 10.3390/s21165576. PMID: 34451018; PMCID: PMC8402252.*

[3] *T. Ngo et al., "Federated Deep Learning for the Diagnosis of Cerebellar Ataxia: Privacy Preservation and Auto-Crafted Feature Extractor," in IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 30, pp. 803-811, 2022, doi: 10.1109/TNSRE.2022.3161272.*

[4] *Alsubai, S.; Alqahtani, A.; Binbusayyis, A.; Sha, M.; Gumaei, A.; Wang, S. Heart Failure Detection Using Instance Quantum Circuit Approach and Traditional Predictive Analysis. Mathematics 2023, 11, 1467. https://doi.org/10.3390/math11061467*

[5] *Shanbehzadeh M, Nopour R, Kazemi-Arpanahi H. Developing an artificial neural network for detecting COVID-19 disease. J Educ Health Promot. 2022 Jan 31;11:2. doi: 10.4103/jehp.jehp_387_21. PMID: 35281397; PMCID: PMC8893090.*

[6] *S. R. Swarna, A. Kumar, P. Dixit and T. V. M. Sairam, "Parkinson's Disease Prediction using Adaptive Quantum Computing," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 1396-1401, doi: 10.1109/ICICV50876.2021.9388628.*

[7] *Zeng, Y.; Wang, H.; He, J.; Huang, Q.; Chang, S. A Multi-Classification Hybrid Quantum Neural Network Using an All-Qubit Multi-Observable Measurement Strategy. Entropy 2022, 24, 394. https://doi.org/10.3390/e24030394*

[8] Abhilasha S, Malay KD (2014) A blind and fragile watermarking scheme for tamper detection of medical images preserving ROI. *IEEE International Conference on Medical Imaging, m- Health and Emerging Communication systems (MedCom)*

[9] *Hsu SY, Yeh LR, Chen TB, Du WC, Huang YH, Twan WH, Lin MC, Hsu YH, Wu YC, Chen HY. Classification of the Multiple Stages of Parkinson's Disease by a Deep Convolution Neural Network Based on 99mTc-TRODAT-1 SPECT Images. Molecules. 2020 Oct 19;25(20):4792. doi: 10.3390/molecules25204792. PMID: 33086589; PMCID: PMC7587595.*

[10] *Yan J, Liao JB, Gao JY, Zhang WW, Huang CM, Yu HL. Fusion of Audio and Vibration Signals for Bearing Fault Diagnosis Based on a Quadratic Convolution Neural Network. Sensors (Basel). 2023 Nov 13;23(22):9155. doi: 10.3390/s23229155. PMID: 38005542; PMCID: PMC10674422.*

[11] *Zhang Z, Mi X, Yang J, Wei X, Liu Y, Yan J, Liu P, Gu X, Yu T. Remote Sensing Image Scene Classification in Hybrid Classical-Quantum Transferring CNN with Small Samples. Sensors (Basel). 2023 Sep 21;23(18):8010. doi: 10.3390/s23188010. PMID: 37766063; PMCID: PMC10537394.*

[12] *Ajlouni N, Özyavaş A, Takaoğlu M, Takaoğlu F, Ajlouni F. Medical image diagnosis based on adaptive Hybrid Quantum CNN. BMC Med Imaging. 2023 Sep 14;23(1):126. doi: 10.1186/s12880-023-01084-5. PMID: 37710188; PMCID: PMC10500912.*

[13] *Baek H, Yun WJ, Park S, Kim J. Stereoscopic scalable quantum convolutional neural networks. Neural Netw. 2023 Aug;165:860-867. doi: 10.1016/j.neunet.2023.06.027. Epub 2023 Jun 28. PMID: 37437364.*

[14] *Liu YJ, Smith A, Knap M, Pollmann F. Model-Independent Learning of Quantum Phases of Matter with Quantum Convolutional Neural Networks. Phys Rev Lett. 2023 Jun 2;130(22):220603. doi: 10.1103/PhysRevLett.130.220603. PMID: 37327416.*

[15] *Wang M, Lee KCM, Chung BMF, Bogaraju SV, Ng HC, Wong JSJ, Shum HC, Tsia KK, So HK. Low-Latency In Situ Image Analytics With FPGA-Based Quantized Convolutional Neural Network. IEEE Trans Neural Netw*