UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Evaluación de Redes Neuronales Convolucionales Cuánticas en la Clasificación de Señales Intrapulso

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Telecomunicación

AUTOR/A: Camacho Sánchez, Miguel

Tutor/a: Colomer Granero, Adrián

Cotutor/a: Naranjo Ornedo, Valeriana

CURSO ACADÉMICO: 2023/2024

Resumen

El cómputo cuántico está avanzando rápidamente, transformando las capacidades computacionales convencionales. Este trabajo se centra en las Redes Neuronales Quanvolucionales (QNNs), evaluando arquitecturas de circuitos cuánticos en un contexto novedoso. Aplicamos estas arquitecturas a la clasificación de modulaciones intrapulso utilizando una base de datos sintética cuidadosamente elaborada. La computación cuántica, con propiedades como la superposición y el entrelazamiento cuántico, ofrece un terreno fértil para repensar enfoques tradicionales en tareas de clasificación. Nuestro estudio proporciona un análisis empírico de varias arquitecturas cuánticas, evaluando su rendimiento en esta tarea específica, comparándolas con modelos clásicos, y destacando sus ventajas y limitaciones actuales. Todos los experimentos se realizan en simuladores cuánticos debido a las limitaciones actuales del hardware cuántico.

Resum

El còmput quàntic està avançant ràpidament, transformant les capacitats computacionals convencionals. Aquest treball se centra en les Xarxes Neuronals Quanvolucionals (QNNs), avaluant arquitectures de circuits quàntics en un context novedós. Apliquem aquestes arquitectures a la classificació de modulacions intrapuls utilitzant una base de dades sintètica acuradament elaborada. La computació quàntica, amb propietats com la superposició i l'enllaç quàntic, ofereix un terreny fèrtil per repensar enfocaments tradicionals en tasques de classificació. El nostre estudi proporciona una anàlisi empírica de diverses arquitectures quàntiques, avaluant el seu rendiment en aquesta tasca específica, comparant-les amb models clàssics, i destacant els seus avantatges i limitacions actuals. Tots els experiments es realitzen en simuladors quàntics a causa de les limitacions actuals del maquinari quàntic.

Abstract

Quantum computing is rapidly advancing, transforming conventional computational capabilities. This work focuses on Quanvolutional Neural Networks (QNNs), evaluating quantum circuit architectures in a novel context. We apply these architectures to the classification of intrapulse modulations using a carefully crafted synthetic database. Quantum computing, with properties such as superposition and entanglement, offers fertile ground for rethinking traditional approaches to classification tasks. Our study provides an empirical analysis of various quantum architectures, assessing their performance in this specific task, comparing them with classical models, and highlighting their current advantages and limitations. All experiments are conducted on quantum simulators due to the current limitations of quantum hardware.

La memoria del TFM del ”EVALUACIÓN DE REDES NEURONALES CONVOLUCIONALES CUÁNTICAS EN LA CLASIFICACIÓN DE SEÑALES INTRAPULSO” debe desarrollar en el texto los siguientes conceptos, debidamente justificados y discutidos, centrados en el ámbito de la COMPUTACION CUANTICA

CONCEPT (ABET)	**CONCEPTO (traducción)**	**¿Cumple? (S/N)**	**¿Dónde? (páginas)**
1. IDENTIFY:	1. IDENTIFICAR:		
1.1. Problem statement and opportunity	1.1. Planteamiento del problema y oportunidad	S	4
1.2. Constraints (standards, codes, needs, requirements & specifications)	1.2. Toma en consideración de los condicionantes (normas técnicas y regulación, necesidades, requisitos y especificaciones)	S	6-19
1.3. Setting of goals	1.3. Establecimiento de objetivos	S	5
2. FORMULATE:	2. FORMULAR:		
2.1. Creative solution generation (analysis)	2.1. Generación de soluciones creativas (análisis)	S	27-34
2.2. Evaluation of multiple solutions and decision-making (synthesis)	2.2. Evaluación de múltiples soluciones y toma de decisiones (síntesis)	S	26
3. SOLVE:	3. RESOLVER:		
3.1. Fulfilment of goals	3.1. Evaluación del cumplimiento de objetivos	S	36-41
3.2. Overall impact and significance (contributions and practical recommendations)	3.2. Evaluación del impacto global y alcance (contribuciones y recomendaciones prácticas)	S	42

Contents

List of Figures

List of Tables

1 Introduction

Convolutional Neural Networks (CNNs) have been a cornerstone in the evolution of machine learning, especially in tasks involving image and video processing. Their unique structure, designed to automatically and adaptively learn spatial hierarchies of features from input data, has been instrumental in various breakthroughs in deep learning. CNNs mimic the human visual system's ability to recognize patterns and objects, making them exceptionally efficient in handling complex visual data. This ability has revolutionized fields such as computer vision, facial recognition, and automated driving systems, setting the stage for advanced machine learning applications and inspiring the exploration of quantum-enhanced models like Quanvolutional Neural Networks (QNNs).

The realm of quantum computing is witnessing a paradigm shift with its integration into conventional neural network architectures, opening new frontiers in computational capabilities. This work spotlights Quanvolutional Neural Networks (QNNs), emphasizing the evaluation of established quantum circuit architectures from the literature in a novel context. We specifically apply these architectures to tackle the complex challenge of intrapulse modulation classification using a meticulously crafted synthetic database

Quantum computing, characterized by its extraordinary computational properties such as quantum superposition and entanglement, provides a fertile ground for rethinking traditional approaches to classification tasks. QNNs, standing at the confluence of quantum computing and neural networks, harness these properties to offer innovative solutions to classification challenges. Our research is dedicated to assessing the practicality and performance of pre-existing quantum architectures when applied to the nuanced task of intrapulse modulation classification. The synthetic dataset employed in this study is designed to closely replicate real-world conditions, providing a robust platform for evaluating the efficacy of QNNs.

Our study primarily focuses on an empirical analysis of various quantum architectures, particularly in their application to a synthetic dataset of intrapulse modulation classification. Instead of advancing theoretical aspects of quantum-enhanced machine learning, our research rigorously tests and compares the performance of different quantum neural network architectures. This approach provides a comprehensive understanding of how these architectures behave under specific machine learning tasks, offering valuable insights into their practical applications and limitations in complex classification scenarios.

In conclusion, our work contributes to the practical assessment of quantum computing principles in neural network technologies. The findings from our investigation offer a grounded perspective on the current capabilities of quantum neural networks.

2 Objectives

The primary objective of this work is to explore the potential of quantum computing in the field of image classification, specifically focusing on intrapulse spectrograms. In particular, we aim to train a quantum circuit to achieve a certain level of performance in classifying these spectrogram images.

Our goals include:

- Designing and implementing a quantum circuit capable of handling image data.
- Training the quantum circuit using a dataset of intrapulse spectrograms.
- Evaluating the performance of the quantum circuit in terms of accuracy and efficiency.
- Comparing the results with classical machine learning approaches to highlight potential advantages and limitations of quantum methods.

Given the current limitations of quantum hardware, all experiments and implementations are conducted using a quantum simulator. This allows us to explore and validate our approach, as real quantum hardware may not yet be capable of executing the necessary parameterized rotational gates with the required precision and stability.

By achieving these objectives, we hope to contribute to the growing body of knowledge in quantum machine learning and demonstrate its applicability to complex tasks like image classification.

3 Background

3.1 Principles of Quantum Mechanics

Quantum mechanics is the branch of physics that describes the behavior of particles at the smallest scales, such as electrons and photons. It operates fundamentally differently from classical mechanics, introducing concepts that seem counterintuitive to our everyday experiences. One of the cornerstone principles is superposition. In classical systems, a bit can be in one of two states: 0 or 1. However, in quantum mechanics, a quantum bit, or qubit, can exist in a superposition of both states simultaneously. This means a qubit can represent both 0 and 1 at the same time, described mathematically by the state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are complex coefficients that represent the probability amplitudes of the qubit being in state 0 or 1.

Another fundamental principle is entanglement [1]. When particles become entangled, the state of one particle is intrinsically linked to the state of another, regardless of the distance separating them. This implies that measuring the state of one entangled particle instantly determines the state of the other, a phenomenon that Einstein famously referred to as "spooky action at a distance." Entanglement is a powerful resource in quantum mechanics, enabling correlations that are impossible in classical systems.

One of the most fundamental and illustrative examples of quantum entanglement involves a pair of qubits. Consider two qubits, each of which can exist in a superposition of the states $|0\rangle$ and $|1\rangle$. When these qubits become entangled, they can form a combined state that cannot be separated into individual qubit states.

A classic example is the Bell state, specifically the $|\Phi^+\rangle$ state. This entangled state is given by:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \tag{2}$$

Here, $|00\rangle$ represents both qubits being in the state $|0\rangle$, and $|11\rangle$ represents both qubits being in the state $|1\rangle$. The Bell state $|\Phi^+\rangle$ is a superposition of these two possibilities.

To understand why this state is entangled, consider the general form of a separable (non-entangled) state for two qubits:

$$|\psi\rangle = (\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \tag{3}$$

Expanding this, we get:

$$|\psi\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle \tag{4}$$

For the state $|\Phi^+\rangle$ to be separable, it must be possible to write it in this form. However, the Bell state $|\Phi^+\rangle$ has only two terms ($|00\rangle$ and $|11\rangle$) with equal coefficients, which cannot be factored into a product of single-qubit states. Specifically, there are no single-qubit coefficients $\alpha_1, \beta_1, \alpha_2, \beta_2$ that can reproduce the equal superposition of $|00\rangle$ and $|11\rangle$.

This inability to factorize the state $|\Phi^+\rangle$ into individual qubit states is what characterizes it as an entangled state. Measuring one qubit immediately affects the state of the other, regardless of the distance between them, demonstrating the quintessential non-local nature of quantum mechanics.

3.2 Bell's Inequality

The Bell inequality, formulated by physicist John Bell in 1964, is a fundamental concept in quantum mechanics that tests the difference between classical and quantum predictions regarding correlations between distant particles.

Classically, under the assumption of local realism, the correlation between measurements on two entangled particles cannot exceed a certain bound. This bound is represented by the value 2. In other words, if the measurements were purely classical and local, the correlation parameter S would satisfy $S \leq 2$.

Quantum mechanically, however, entangled particles can exhibit stronger correlations that violate the classical bound predicted by local realism. Specifically, quantum theory predicts that the maximum value of the correlation parameter S can reach $2\sqrt{2}$.

In the context of the Bell inequality, local realism refers to the idea that physical properties of objects exist independently of observation (realism) and that distant objects cannot instantaneously influence each other (locality). The violation of the Bell inequality in quantum experiments implies that at least one of these assumptions—locality or realism—is not valid in the quantum realm.

Experimental tests of the Bell inequality have consistently shown results where the measured correlation parameter S exceeds the classical limit of 2, often approaching $2\sqrt{2}$ [2]. This violation of the Bell inequality strongly suggests that the correlations observed between entangled particles cannot be explained by classical physics alone. Instead, it indicates the existence of quantum entanglement, where particles can instantaneously influence each other regardless of the distance between them, thus confirming the non-local nature of quantum mechanics.

3.3 Principles of Quantum Computing

Building on these quantum mechanical principles, quantum computing leverages the unique properties of qubits to perform computations in ways that classical computers cannot. In classical computing, information is processed using bits that are strictly 0 or 1. Quantum computing, however, uses qubits that utilize superposition and entanglement to process information [3].

A quantum computer can perform many calculations simultaneously due to the superposition of qubits. This parallelism allows quantum computers to tackle complex problems more efficiently than classical computers. For example, in a system with n qubits, a quantum computer can represent and manipulate 2^n states simultaneously, providing an exponential increase in computational power.

Entanglement further enhances this power by allowing qubits to be interdependent. Quantum gates, the building blocks of quantum circuits, manipulate qubits in ways that can create and utilize entanglement, enabling complex operations that are infeasible for classical computers. These gates are analogous to classical logic gates but operate on qubits through quantum operations.

Moreover, quantum algorithms are specifically designed to exploit quantum mechanical phenomena. Shor's algorithm, for instance, can factor large numbers exponentially faster than the best-known classical algorithms, posing a significant challenge to classical encryption methods. Similarly, Grover's algorithm provides a quadratic speedup for unstructured search problems, showcasing the potential of quantum computing in diverse applications.

In summary, quantum computing is built on the principles of superposition and entanglement, utilizing these phenomena to perform computations in fundamentally new and powerful ways. As research advances, quantum computing holds the promise of revolutionizing fields from cryptography to material science, offering solutions to problems that are currently intractable for classical computers.

Bloch Sphere

The Bloch sphere is a fundamental concept in quantum mechanics, providing a geometric representation of the state of a single qubit in quantum computing [4].

The Bloch sphere, see Figure 1, maps these quantum states onto the surface of a sphere. The north pole represents the state $|0\rangle$, where the qubit is certain to be measured as 0. The south pole represents the state $|1\rangle$, where the qubit is certain to be measured as 1. Points on the surface of the sphere between these poles represent superpositions of $|0\rangle$ and $|1\rangle$, where the qubit has probabilities of being measured as 0 or 1.

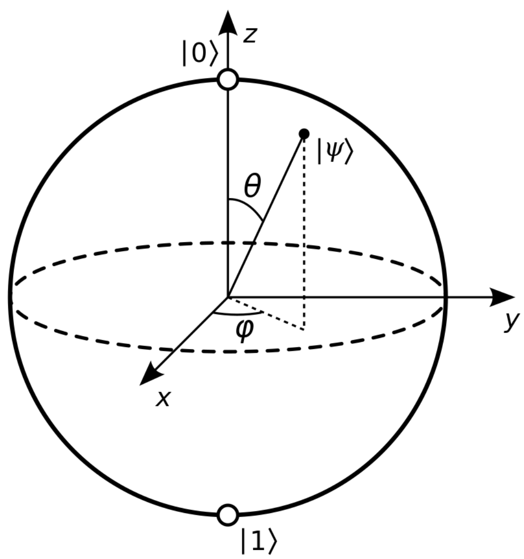


Figure 1: Bloch Sphere Representation

Quantum operations, such as rotations and transformations applied to qubits, are represented as rotations of the Bloch vector around different axes of the sphere. These operations are fundamental in quantum algorithms, allowing manipulation of qubit states to perform computations.

The Bloch sphere not only serves as a visualization tool but also helps in understanding the behavior of qubits under various quantum operations. It illustrates how quantum gates, like the Hadamard gate or phase gates, affect the qubit's state by altering its position on the sphere.

3.3.1 Quantum Gates

In classical computing, gates manipulate bits. Similarly, in quantum computing, quantum gates manipulate qubits. However, unlike classical gates, quantum gates can create superpositions and entanglements due to the principles of quantum mechanics. Here are some of the most important quantum gates:

- **Pauli-X Gate**
  The Pauli-X gate is the quantum equivalent of the classical NOT gate. It flips the state of a qubit:

  $$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{5}$$

  If a qubit is in state $|0\rangle$, applying the X gate will change it to $|1\rangle$, and vice versa.

- **Pauli-Y Gate**
  The Pauli-Y gate introduces a phase flip combined with a bit flip:

  $$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{6}$$

  This gate changes the state of the qubit and multiplies it by the imaginary unit i.

- **Pauli-Z Gate**
  The Pauli-Z gate applies a phase flip to the qubit:

  $$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{7}$$

  This gate leaves $|0\rangle$ unchanged but multiplies the $|1\rangle$ state by -1.

- **Hadamard Gate**
  The Hadamard gate creates superpositions by transforming the basis states:
  $$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{8}$$
  Applying the Hadamard gate to $|0\rangle$ creates an equal superposition of $|0\rangle$ and $|1\rangle$:
  $$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{9}$$

- **Controlled-NOT (CNOT) Gate**
  The CNOT gate operates on two qubits, flipping the state of the second qubit (target) if the first qubit (control) is in the state $|1\rangle$:
  $$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{10}$$

- **Toffoli Gate**
  The Toffoli gate, or CCNOT gate, is a three-qubit gate that flips the state of the third qubit (target) if the first two qubits (controls) are in the state $|1\rangle$:
  $$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{11}$$

3.4 Current State of Quantum Computers

The current state of quantum computers is in a developmental phase known as the Noisy Intermediate-Scale Quantum (NISQ) era. NISQ devices have a limited number of qubits and are subject to significant errors due to noise and imperfections in quantum operations. Although they have shown great potential, they are not yet capable of reliably performing complex calculations on a large scale [5].

Currently, one of the main challenges in quantum computing is maintaining qubit coherence and minimizing gate errors. Quantum systems are highly sensitive to their environment, and even slight disturbances can cause decoherence, leading to errors in computations. Additionally, implementing error correction codes in quantum systems is still a significant hurdle due to the overhead required in terms of additional qubits and operations.

Given these limitations, many quantum algorithms and experiments, including those presented in this work, are executed using quantum simulators rather than actual quantum hardware. Simulators allow researchers to model and test quantum circuits with high fidelity, providing insights into their behavior and performance without the noise and errors inherent in current quantum hardware. Specifically, for this work, all experiments involving parameterized rotational gates were conducted on simulators, as these operations are not yet feasible on current quantum computers.

All quantum circuits in this study were implemented using the PennyLane library in Python. PennyLane provides a powerful framework for quantum machine learning and quantum computing simulations, allowing researchers to design, optimize, and evaluate quantum algorithms using various quantum simulators and hardware platforms seamlessly.

In summary, while quantum computers hold immense promise, the technology is still in its early stages. Simulations play a crucial role in advancing our understanding and development of quantum algorithms, paving the way for future breakthroughs when more advanced and reliable quantum hardware becomes available.

3.5 Measurement and Expected Value in Quantum Computing

In quantum computing, measurement plays a fundamental role in extracting information from quantum systems. Unlike classical systems where measurements yield definite outcomes, quantum measurements are probabilistic due to the principles of superposition and uncertainty inherent in quantum mechanics.

3.5.1 Measurement in Quantum Computing

Measurement in quantum computing refers to the process of determining the state of a quantum system by performing a physical interaction that collapses the quantum state into one of its basis states [6]. For a qubit, which can exist in a superposition of states $\alpha|0\rangle + \beta|1\rangle$, measuring collapses the qubit into either state $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$, respectively. This probabilistic nature arises from the Born rule in quantum mechanics, which relates the probability of measurement outcomes to the magnitudes of the probability amplitudes α and β.

In the context of the Bloch sphere representation, measurement can be understood as projecting the quantum state vector onto one of the basis vectors, analogous to taking the inner product or scalar projection of the quantum state vector onto the measurement basis vector. This projection is influenced by the quantum gates applied to the qubits during computation, such as controlled rotations, which manipulate the quantum state and determine the probabilities of measurement outcomes. Consequently, the expected values obtained from measurement serve as the class labels for the input data in quantum classification tasks.

3.5.2 Expected Value in Quantum Computing

The expected value in quantum computing refers to the average outcome of a measurement performed on a quantum state. It is computed as the sum of all possible measurement outcomes weighted by their probabilities. Mathematically, the expected value $\langle \hat{O} \rangle$ of an observable $\hat{O}$ in the quantum state $|\psi\rangle$ is given by:

$$\langle \hat{O} \rangle = \langle \psi | \hat{O} | \psi \rangle \tag{12}$$

where $\langle \psi |$ denotes the bra vector conjugate of $|\psi\rangle$. The operator $\hat{O}$ represents the observable quantity being measured.

In quantum algorithms, such as those used in quantum machine learning or optimization tasks, computing the expected value of observables plays a crucial role. It allows researchers to extract meaningful information about the quantum state and to make decisions based on the outcomes of measurements.

Measurement and expected value calculations are essential components of quantum algorithms and protocols, enabling quantum computers to process information and provide insights into complex systems that classical computers may struggle to analyze efficiently.

Let's consider a specific example to illustrate this concept. Suppose we have a qubit initially in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers (probability amplitudes) such that $|\alpha|^2 + |\beta|^2 = 1$. The observable we want to measure is the Pauli-Z operator Z, represented in equation 7 as:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

To find the expected value $\langle Z \rangle = \langle \psi|Z|\psi\rangle$, we perform the following calculations:

1. Compute $\langle \psi|$ (bra vector conjugate of $|\psi\rangle$):

$$\langle \psi| = (\alpha^*, \beta^*) \tag{13}$$

where α^* and β^* are the complex conjugates of α and β, respectively.

2. Compute the product $Z|\psi\rangle$:

$$Z|\psi\rangle = Z(\alpha|0\rangle + \beta|1\rangle) = \alpha Z|0\rangle + \beta Z|1\rangle \tag{14}$$

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle \tag{15}$$

$$Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle \tag{16}$$

3. Compute the inner product $\langle \psi|Z|\psi\rangle$:

$$\langle \psi|Z|\psi\rangle = (\alpha^*, \beta^*) \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{17}$$

Performing the matrix multiplication:

$$\langle \psi|Z|\psi\rangle = \begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \begin{pmatrix} \alpha \\ -\beta \end{pmatrix} \tag{18}$$

$$\langle \psi|Z|\psi\rangle = \alpha^*\alpha + \beta^*\beta(-1) \tag{19}$$

$$\langle \psi|Z|\psi\rangle = |\alpha|^2 - |\beta|^2 \tag{20}$$

Thus, the expected value $\langle Z \rangle$ represents the difference in probabilities of measuring state $|0\rangle$ and state $|1\rangle$ in the quantum state $|\psi\rangle$, reflecting the superposition of $|0\rangle$ and $|1\rangle$ and the probabilities $|\alpha|^2$ and $|\beta|^2$ associated with each state.

3.6 Background of Machine Learning: From Perceptron to Convolutional Neural Networks (CNNs)

Machine learning (ML) has evolved dramatically over the decades, beginning with simple models and advancing to complex architectures that power today's AI systems. This journey can be traced from the development of the perceptron to the sophisticated convolutional neural networks (CNNs) used extensively in image and video recognition tasks.

3.6.1 The Perceptron: The Birth of Machine Learning

The perceptron, introduced by Frank Rosenblatt in 1958, is considered one of the earliest models of machine learning. It is a type of linear classifier that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. Essentially, a perceptron takes several binary inputs, processes them through weighted sums, and passes the result through an activation function (typically a step function) to produce a binary output.

The simplicity of the perceptron model is both its strength and its limitation. While it laid the foundation for later developments in ML, it could only solve linearly separable problems. This limitation was highlighted by Minsky and Papert in their book *Perceptrons* (1969), which slowed down research in neural networks for several years.

3.6.2 Multilayer Perceptron (MLP): Introduction of Non-linearity

To overcome the limitations of the single-layer perceptron, researchers developed the multilayer perceptron (MLP). An MLP consists of multiple layers of perceptrons (neurons), including an input layer, one or more hidden layers, and an output layer. The hidden layers enable the network to learn and represent complex non-linear relationships in the data [7].

Each neuron in an MLP uses a non-linear activation function, such as the sigmoid or tanh function, which allows the network to approximate any continuous function. The introduction of backpropagation, a method for training MLPs by efficiently computing gradients, further enhanced their capability and popularity. Backpropagation, proposed by Rumelhart, Hinton, and Williams in 1986, enables the adjustment of weights in the network by minimizing the error between predicted and actual outputs.

3.6.3 Evolution to Deep Learning

As computational power and data availability increased, so did the depth of neural networks. Deep learning, a subset of ML, involves training neural networks with many layers, which can learn representations at multiple levels of abstraction. Deep learning models have revolutionized various fields, from natural language processing to computer vision, due to their ability to automatically learn intricate features from raw data [8].

3.6.4 Convolutional Neural Networks (CNNs): A Leap in Image Processing

Convolutional Neural Networks (CNNs) are a specialized kind of deep learning model designed to process data with a grid-like topology, such as images. Introduced by Yann LeCun and his collaborators in the late 1980s and early 1990s, CNNs are particularly effective for image recognition and classification tasks [9].

A CNN architecture typically includes several types of layers:

- **Convolutional Layers**: These layers apply a set of filters to the input data to create feature maps. Each filter detects different features such as edges, textures, or patterns. The convolution operation helps in preserving the spatial relationship between pixels by learning image features using small squares of input data.

- **Pooling Layers**: Pooling layers reduce the dimensionality of the feature maps, which decreases the computational load and helps in making the detection of features invariant to small translations of the input.

- **Fully Connected Layers**: After several convolutional and pooling layers, the high-level reasoning is done via fully connected layers, which have connections to all activations in the previous layer, akin to traditional MLPs.

- **Activation Functions**: Non-linear activation functions such as ReLU (Rectified Linear Unit) are used after convolutional and fully connected layers to introduce non-linearity into the model, allowing it to learn more complex functions.

CNNs have demonstrated outstanding performance in various applications. Notably, in 2012, a CNN-based model called AlexNet won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a significant margin, which sparked widespread interest and further research in deep learning and CNNs.

In summary, machine learning has progressed from the simple perceptron to complex deep learning models. Each advancement has built upon the last, leading to the powerful convolutional neural networks that are integral to modern AI systems, particularly in the field of image and video analysis.

3.6.5 Loss Functions in Machine Learning

In machine learning, loss functions play a crucial role in the training process of models. They measure how well the model's predictions match the actual data, guiding the optimization process to minimize errors and improve performance [10]. Two commonly used loss functions are Mean Squared Error (MSE) and Cross-Entropy Loss.

The Mean Squared Error (MSE) is a loss function primarily used for regression tasks. However, it can also be applied to classification problems, especially when the output is a continuous value rather than a discrete class label. It calculates the average squared difference between the predicted values and the actual values. The MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{21}$$

where y_i represents the actual values, $\hat{y}_i$ represents the predicted values, and n is the number of data points. The MSE penalizes larger errors more significantly than smaller ones due to the squaring of the differences, making it sensitive to outliers.

Cross-Entropy Loss, also known as log loss, is widely used for classification tasks. It measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-Entropy Loss increases as the predicted probability diverges from the actual label. The formula for Cross-Entropy Loss is:

$$\text{Cross-Entropy Loss} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C} y_{i,c}\log(\hat{y}_{i,c}) \tag{22}$$

where $y_{i,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for observation i, $\hat{y}_{i,c}$ is the predicted probability that observation i is of class c, and C is the total number of classes. Cross-Entropy Loss effectively penalizes the model when it assigns low probabilities to the correct class labels, thus encouraging the model to improve its probability estimates.

In summary, loss functions are essential components in machine learning that quantify the difference between the predicted outputs and the true labels. MSE is primarily useful for regression problems but can also be applied to classification tasks, while Cross-Entropy Loss is preferred for classification tasks. Both loss functions guide the optimization process to improve the model's accuracy and generalization capabilities.

3.7 Quantum Machine Learning

Quantum Machine Learning (QML) represents the intersection of quantum computing and classical machine learning. It aims to leverage the principles of quantum mechanics to enhance the performance of machine learning algorithms. The potential of QML lies in its ability to process and analyze large datasets more efficiently than classical computers by exploiting quantum phenomena such as superposition, entanglement, and quantum parallelism [11].

3.7.1 Variational Quantum Circuits (VQCs)

At the heart of many QML algorithms are Variational Quantum Circuits (VQCs), also known as parameterized quantum circuits. These circuits are composed of a sequence of quantum gates whose parameters can be tuned to optimize a certain objective function. The general workflow involves initializing the parameters, running the quantum circuit, measuring the outcome, and then using classical optimization techniques to adjust the parameters to minimize or maximize the objective function.

VQCs are particularly powerful because they combine the strengths of both quantum and classical computing. The quantum part of the computation can potentially explore the solution space more efficiently, while the classical part handles the optimization, making the process manageable even with current noisy intermediate-scale quantum (NISQ) devices.

3.7.2 Quantum Gates and Rotation Gates

Quantum gates are the building blocks of quantum circuits. They are analogous to classical logic gates but operate on quantum bits (qubits). Some of the most important quantum gates in VQCs are the rotation gates. These gates perform rotations on the state of a single qubit around a specified axis of the Bloch sphere.

- **RX(θ) Gate**: Rotates the qubit state around the X-axis by an angle θ.
- **RY(θ) Gate**: Rotates the qubit state around the Y-axis by an angle θ.
- **RZ(θ) Gate**: Rotates the qubit state around the Z-axis by an angle θ.

These gates are parameterized by the angle of rotation, and adjusting these angles allows for the fine-tuning of the quantum state during the optimization process.

3.7.3 Parameter-Shift Rule

To optimize the parameters in a VQC, we need to compute the gradients of the objective function with respect to the circuit parameters. One of the methods used for this purpose in QML is the parameter-shift rule. This rule provides an efficient way to calculate the derivative of the expectation value of an observable with respect to a parameter of a quantum gate.

The parameter-shift rule for a parameterized gate $U(\theta)$ (such as a rotation gate) is given by:

$$\frac{\partial\langle\hat{O}\rangle}{\partial\theta} = \frac{\langle\hat{O}\rangle(\theta+\frac{\pi}{2}) - \langle\hat{O}\rangle(\theta-\frac{\pi}{2})}{2} \tag{23}$$

where $\langle\hat{O}\rangle(\theta)$ is the expectation value of the observable $\hat{O}$ with the parameter θ.

This rule works by evaluating the quantum circuit at two shifted parameter values, $\theta+\frac{\pi}{2}$ and $\theta-\frac{\pi}{2}$, and then combining these results to obtain the gradient. This approach is particularly advantageous in quantum computing because it avoids the need for finite-difference approximations, which can be numerically unstable.

Quantum Machine Learning is an emerging field that seeks to harness the power of quantum computing to solve complex machine learning problems. By using variational quantum circuits, rotation gates, and the parameter-shift rule, QML algorithms can efficiently explore and optimize high-dimensional solution spaces. As quantum technology advances, QML holds the promise of transforming various domains by providing unprecedented computational power and speed.

4 State of the Art

The landscape of machine learning has been profoundly transformed by the advent and evolution of Convolutional Neural Networks (CNNs). Since their inception in the 1990s, CNNs have become a staple in deep learning due to their unique ability to automatically and adaptively learn spatial hierarchies of features from input data. This innovation, drawing inspiration from the human visual system, has spearheaded advances in complex tasks such as image and video recognition, computer vision, and facial recognition, setting a benchmark in the field.

The problem of recognizing intrapulse signal modulation using Convolutional Neural Networks (CNNs) and spectrograms has been extensively studied, yielding promising results. Various research efforts have demonstrated the effectiveness of CNNs in accurately classifying different types of modulation schemes from radar signals [12]. These studies utilize spectrograms to convert time-domain radar signals into the frequency domain, allowing CNNs to exploit their strong feature extraction capabilities. The combination of CNNs and spectrograms has shown high accuracy and robustness in modulation classification tasks, highlighting its potential as a powerful tool in radar signal processing.

Advancements in quantum computing have opened new views in machine learning, leading to the development of Quantum Convolutional Neural Networks (QCNNs). The pioneering work by Cong, Choi, and Lukin in 2019 [13] marked a significant step in this direction. Their QCNN model, designed for binary classification tasks, utilized the principles of quantum computing to efficiently handle input sizes of N qubits. This model showcased the potential of QCNNs in accurately recognizing quantum states associated with a

1D symmetry-protected topological phase, demonstrating an edge over existing approaches. The QCNN model is based on the Multi-scale Entanglement Renormalization Ansatz (MERA), a tensor network structure that effectively captures entanglement at multiple scales. Cong and his team innovatively used a reverse MERA structure, which processes information in a hierarchical manner, similar to how traditional convolutional neural networks downsample and then upsample data. This reverse MERA approach enables the QCNN to efficiently compress and then reconstruct quantum information, leading to enhanced performance in quantum state recognition tasks.

In a subsequent development, Henderson presented in 2020 a hybrid model that combines the advantages of classical neural networks and the unique capabilities of quantum circuits [14]. This model, referred to as Quanvolutional Neural Networks (QNNs), integrates quanvolutional layers within the traditional CNN architecture. These layers use random quantum circuits to locally transform subsections of the input data, generating feature maps that are integrated with the classical layers. This hybrid approach not only improves the model's accuracy in classification tasks but also accelerates the training process, thus demonstrating a promising path for the use of quantum computers in the near term, known as NISQ (Noisy Intermediate-Scale Quantum) devices. In the context of Quantum Convolutional Neural Networks (QCNNs), the term *hybrid* refers to the integration of both quantum and classical computing elements within a single machine learning model. This integration can be realized in two primary ways: by combining quantum and classical layers or by using quantum layers exclusively while employing classical methods for gradient calculation. Here, Henderson use the first aproach.

Building on this foundation, subsequent studies elaborated by Chalumuri et al. (2021)[15] and Bokhan et al.(2023) [16] extended the application of QCNNs to multiclass classification problems. Chalumuri and colleagues proposed a hybrid model using only quantum layers and observed improved performance by incorporating ancillas. Ancillas are auxiliary qubits used to encode additional information, enhanced the model's ability to handle the complexity of multiclass classification.

Meanwhile, Bokhan and his team proposed a robust model also employing ancillas, but they added Toffoli gates at the end of the circuit. This approach allowed Bokhan's model to achieve better performance compared to Chalumuri's model.

Recent studies have also focused on evaluating the performance of these quantum convolutional networks by integrating both quantum and classical layers, particularly on datasets like MNIST, as discussed in [17].

Our research contributes to this evolving landscape by applying QCNNs to a synthetic multiclass dataset of intrapulse signals. While previous works have primarily tested their models on the MNIST dataset, we extend the evaluation to a different dataset to explore the versatility of QCNNs. We focus on empirically analyzing various quantum architectures, assessing their performance in this specific classification context. Our study, while not advancing theoretical aspects of quantum-enhanced machine learning, provides a critical evaluation of existing QCNN models, offering insights into their practical applications in sophisticated classification scenarios.

5 Data

5.1 Our Dataset

Intrapulse modulations are sophisticated variations within a radar pulse, where the modulation can involve changes in frequency, phase, or amplitude according to a specific, predefined pattern. These modulations are critical in advanced radar systems as they enhance the system's ability to detect and characterize targets with higher resolution and accuracy. A spectrogram, which is a visual representation of the spectrum of frequencies of a signal as they vary with time, serves as an essential tool in analyzing these modulations. It provides a two-dimensional depiction of the frequency spectrum over the pulse duration, highlighting the temporal evolution of the signal's frequency content.

For this study, a simulator was utilized to generate a comprehensive synthetic database of intrapulse modulations. The simulation parameters were meticulously chosen to reflect realistic operational conditions. The sampling frequency (f_s) was set to 50 MHz to adequately capture the frequency dynamics of the modulations. The pulse width (p_s) was configured at 50 microseconds (μs), a typical setting for many radar applications, allowing for sufficient data collection without excessive signal loss. The signal-to-noise ratio (SNR) ranged from -6 to 10 dB in 2 dB increments, providing a spectrum of conditions from low to relatively high signal quality.

These modulations techniques are crucial in modern radar systems for several reasons, enhancing the radar's capabilities in various operational environments. The types of modulations choosen for this study were this four, represented in Figure 2:

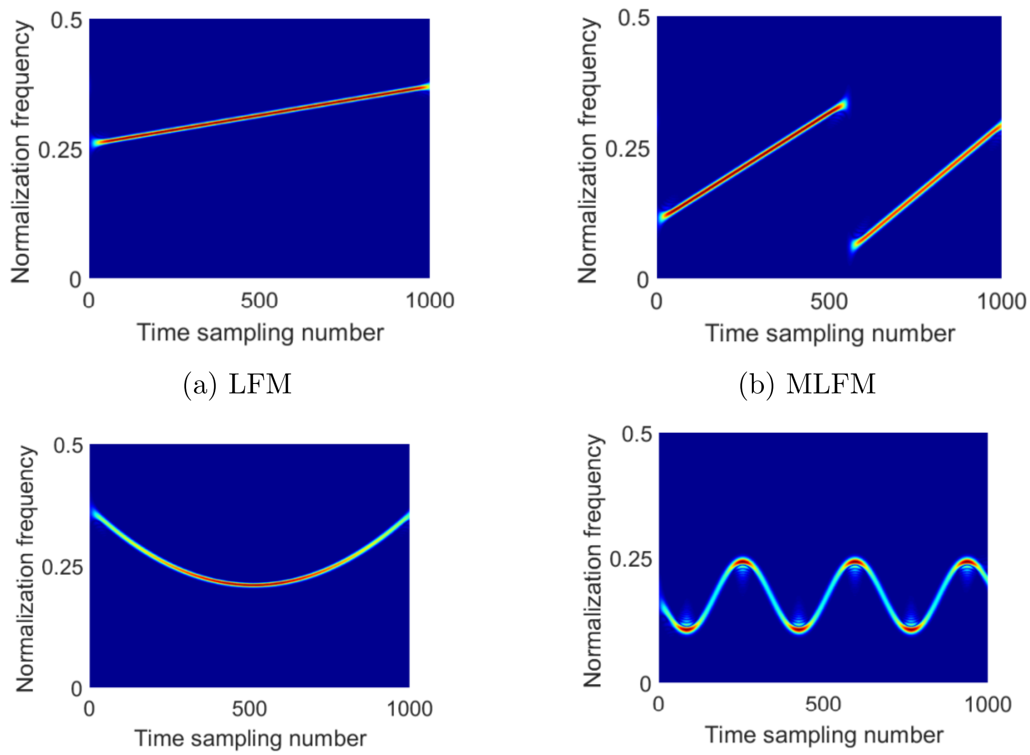
- **Linear Frequency Modulation (LFM)**: Linear Frequency Modulation (LFM) involves varying the frequency of the radar signal linearly over the pulse duration. Also known as chirp modulation, LFM improves the radar's range resolution and allows for better discrimination between targets at different distances. By linearly sweeping the frequency, it spreads the signal's energy over a broader bandwidth, which enhances the ability to resolve targets that are close together in range. This type of modulation is particularly beneficial in environments where high resolution is necessary to distinguish between closely spaced objects.

- **Multi-Linear Frequency Modulation (MLFM)**: Multi-Linear Frequency Modulation (MLFM) extends the concept of LFM by incorporating multiple linear frequency sweeps within a single pulse. This approach allows for even finer resolution and better clutter rejection. MLFM is effective in complex environments where multiple targets may be present, as it helps in distinguishing between different objects by providing a more detailed frequency signature.

- **Quadratic Frequency Modulation (QFM)**: Quadratic Frequency Modulation (QFM) involves a quadratic variation of the frequency over the pulse duration. Unlike LFM, which varies the frequency linearly, QFM uses a quadratic relationship, which can provide different resolution and sidelobe characteristics. This type of modulation is useful in scenarios where specific range and Doppler resolution characteristics are required. QFM can also help in reducing the impact of range sidelobes, improving target detection performance.

- **Sinusoidal Frequency Modulation (SFM)**: Sinusoidal Frequency Modulation (SFM) modulates the frequency of the radar signal in a sinusoidal pattern within the pulse. This type of modulation is beneficial for certain types of signal processing techniques, such as those that exploit the periodic nature of the sinusoidal frequency variation. SFM can be used to enhance the detection of certain types of targets or to improve the radar's resistance to certain types of interference and jamming.

For each type of modulation, 1800 unique spectrograms of size 16x16 pixels were generated, culminating in a total of 9000 images. The dataset was systematically divided into training (70%), validation (10%), and testing (20%) subsets. This structured partitioning facilitates rigorous model training and evaluation, ensuring that the quantum convolutional neural networks developed can be thoroughly assessed across a variety of signal conditions.



(a) LFM (b) MLFM

(c) QFM (d) SFM

Figure 2: Different types of intrapulse signal modulations used to train the models[12]. (a) LFM, (b) MLFM, (c) QFM, and (d) SFM.

5.2 Data Preprocessing and Quantum Encoding

The preprocessing of spectrograms in our study involves several crucial steps to ensure the data is suitable for quantum computing applications. Initially, each spectrogram undergoes normalization to scale the pixel values to a consistent range, typically between 0 and 1. This step is vital for maintaining numerical stability and enhancing the effectiveness of the quantum models.

After normalization, each spectrogram undergoes a flattening process, where the 2-dimensional spectrogram matrix is reshaped into a 1-dimensional vector. In our specific implementation, this results in an input vector of size 256. By flattening the spectrograms, we preserve the essential features extracted during preprocessing while transforming the data into a format that can be efficiently processed by both classical and quantum machine learning models.

Following normalization, encoding in quantum computing refers to the process of translating classical information into a format that can be processed by a quantum computer. This is a fundamental step as it allows quantum algorithms to leverage the inherent advantages of quantum mechanics, such as superposition and entanglement. The choice of encoding method can significantly influence the efficiency and success of quantum computations.

In the process of quantum encoding, a mapping from classical data to a quantum state is performed through a unitary transformation $U_\phi(x)$. This mapping is expressed as $x \in X \to |\phi(x)\rangle \in H$, which is equivalent to applying a unitary transformation $U_\phi(x)$ to the initial state $|0\rangle^n$, where n is the number of qubits.

Various techniques of quantum encoding offer specific approaches, as detailed in the article "Quantum convolutional neural network for classical data classification" by Hur, Kim, and Park (2022) [18]. In our study, the quantum encoding technique employed for both quantum models is amplitude embedding. This technique is particularly efficient for encoding high-dimensional data into quantum circuits, leveraging the principle of superposition to represent a large amount of information within a quantum state. In amplitude embedding, the classical data vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is normalized and embedded into the quantum state $|\psi\rangle$ such that:

$$|\psi\rangle = \sum_{i=1}^{n} x_i |i\rangle, \tag{24}$$

where x_i are the components of the classical data vector and $|i\rangle$ represent the basis states of the quantum system. This encoding process ensures that the entire classical dataset can be efficiently transformed into a quantum state, facilitating subsequent quantum operations and computations.

6 Methodology

In this study, we explored three distinct architectures to evaluate their performance on a synthetic dataset for intrapulse modulation classification. The architectures include two fully quantum models and one classical model. The quantum models employ purely quantum convolutional neural networks (QCNNs), designed to leverage the unique computational properties of quantum mechanics. The classical model, a conventional convolutional neural network (CNN), serves as a benchmark for comparison. Detailed descriptions of these methodologies and their implementation will be provided in the following sections.

Given that the quantum circuits require the input spectrogram images to be flattened into one-dimensional vectors, resulting in the loss of spatial information, we have chosen to compare the performance of the quantum convolutional neural networks (QCNNs) with a multilayer perceptron (MLP) instead of a conventional convolutional neural network (CNN). This comparison is considered fairer as the MLP, like the quantum models, does not inherently leverage spatial relationships in the data.

The workflow for the quantum circuits involves several key steps. First, the spectrogram image is flattened into a one-dimensional vector. This vector is then encoded into a quantum state using appropriate quantum encoding techniques. Once encoded, the quantum state is processed through the quantum circuit. The expectation values of the ancilla qubits are measured to match the target class in a one-hot encoding scheme, thereby facilitating the classification task.

In contrast, the workflow for the classical model follows a similar pattern but without the encoding and expectation value measurement phases. The spectrogram image is flattened into a one-dimensional vector and passed directly through the classical neural network. The output probabilities obtained from the network are then used to determine the class.

This approach allows us to compare the performance of quantum and classical models on the same classification task, providing insights into the advantages or disadvantages of quantum computing for machine learning applications.

6.1 Description of the First Approach: Cong's Quantum Convolutional Neural Network (QCNN) Model

The Quantum Convolutional Neural Network (QCNN) proposed by Cong, Choi, and Lukin represents a pioneering approach in the realm of quantum machine learning, particularly designed to leverage the principles of quantum computing for enhanced performance in complex classification tasks. This model extends the classical Convolutional Neural Network (CNN) framework into the quantum domain, offering novel mechanisms for data processing and pattern recognition through quantum circuits. A general scheme is presented in Figure 3.

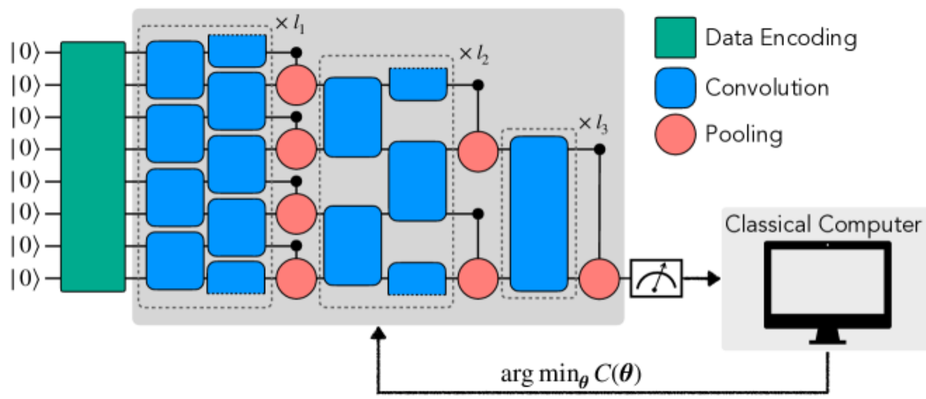


Figure 3: General scheme for Quatum Convolutional Neural Networks [18]

The final architecture implemented, represented in Figure 4, consists of 8 input wires and 4 ancilla wires. Each input wire corresponds to a qubit that encodes the quantum state representation of an input image. Given that we are using amplitude encoding for images of size 16×16, which require 256-dimensional vectors, 8 qubits are employed to handle this high-dimensional input data efficiently.

Additionally, there are 4 ancilla wires integrated into the circuit design, corresponding to the number of classes in our classification task. Ancilla qubits do not directly encode input data but interact with the data qubits during quantum processing to aid in distinguishing between the different classes. Their role is crucial in enhancing the measurement outcomes and enabling multi-class classification capabilities within the quantum circuit.

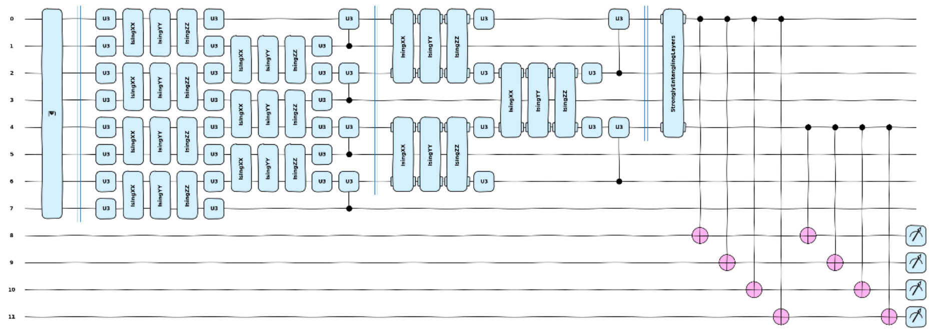


Figure 4: Architecture implemented in Pennylane

Key Components

1. **Quantum Convolution Layer:** The quantum convolution layer operates similarly to its classical counterpart by detecting patterns in input data, but it uses quantum gates to achieve this. These gates are applied to pairs of adjacent qubits, transforming their state in a manner analogous to how classical filters work on pixel values. This quantum operation helps in identifying hidden states and extracting relevant features from the input data, see Figure 5.

   In our approach, convolutional layers have been implemented using the U3 gate and Ising gates. The U3 gate is a fundamental single-qubit gate in quantum computing, characterized by three parameters that allow for arbitrary rotations of the qubit state around the Bloch sphere. These parameters control the amplitudes and phases of the quantum state, enabling precise manipulation and computation within quantum circuits.

   On the other hand, Ising gates in quantum computing refer to gates that implement interactions similar to the Ising model from classical physics. These gates are pivotal in quantum annealing and optimization algorithms, where they model spin interactions between qubits. Ising gates are particularly useful in quantum algorithms for addressing combinatorial optimization problems and for implementing variational quantum algorithms.

   The utilization of U3 gates for rotational operations and Ising gates for modeling interactions between qubits enables our quantum architecture

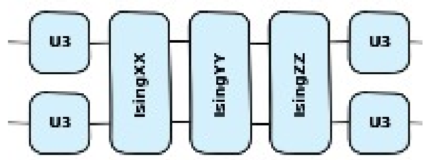


Figure 5: Gates used for the convolutional layer in the first approach

to perform both precise single-qubit rotations and effective pairwise qubit interactions. This combination supports the implementation of quantum convolutional layers capable of extracting features and performing computations crucial for quantum machine learning tasks.

2. **Quantum Pooling Layer:** The quantum pooling layer aims to reduce the system's dimensionality by selectively observing qubits or applying specific quantum gates, such as controlled NOT (CNOT) gates. This reduction process helps manage the data complexity and prevents overfitting, which is crucial for the scalability and efficiency of the model. By reducing the number of qubits while preserving essential information, the pooling layer mimics the down-sampling process in classical CNNs.

   In our case, we have implemented the pooling operation using a controlled-U3 gate. This means that the U3 gate, is controlled by another qubit. In quantum computing, a controlled gate operates on two qubits, where the target qubit (in this case, the qubit undergoing the U3 gate operation) is only rotated if the control qubit satisfies a specific condition, usually being in the state $|1\rangle$. This controlled mechanism allows for conditional operations based on the state of the control qubit, providing a way to selectively apply transformations and manage information flow within the quantum pooling layer.

3. **Fully Connected Layer:** After the data has been processed and reduced through the convolution and pooling layers, the fully connected layer is responsible for the final classification task. This layer uses a quantum circuit to process the extracted features and predict the output class, similar to the final layers in classical neural networks.

   In our implementation, we utilized the PennyLane library's *Strongly Entangling Layers* module for the fully connected layer. This module implements a sequence of alternating layers of single-qubit rotations and entangling gates, such as CNOT gates, across all qubits in the circuit. The entangling gates ensure that each qubit becomes entangled with every other qubit in the circuit, fostering strong correlations across the quantum state space.

4. **Integration of Ancillas:** In our endeavor to achieve multi-class classification using quantum circuits, the integration of ancillas plays a crucial role. Ancillas are additional qubits not directly encoding input data, interact with data qubits during quantum operations. They facilitate the manipulation of quantum states and enable measurements that can differentiate between multiple classes more effectively than using only data qubits.

   The integration of ancillas into quantum circuits for multi-class classification is achieved through Controlled-NOT (CNOT) gates, as described in [15], see Figure 6. These gates entangle ancillas with data qubits, allowing ancillas to influence and interact with the quantum state during computation. This integration enhances the quantum classifier's ability to classify multiple classes accurately.

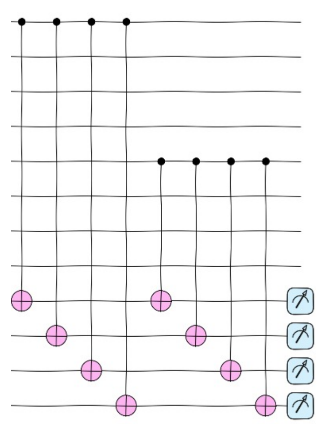


Figure 6: Ancillas mechanism used in the first approach

The QCNN offers several advantages: it operates with logarithmic depth ($O(\log(n))$), making it significantly more efficient than classical CNNs, which require quadratic operations; its design is well-suited for implementation on current small-scale quantum computers and is expected to scale with advancements in quantum computing technology; and it incorporates Quantum

Error Correction (QEC) techniques to handle errors inherent in quantum computing, enhancing the model's reliability and accuracy.

6.2 Description of the Second Approach: Bokhan‘s Architecture

The second model employs a quantum architecture that utilizes Toffoli gates, which are essential for implementing controlled-controlled operations in quantum computation. We use the architecture proposed by Bokhan, Mastiukova, Boev, Trubnikov, and Fedorov (2022) [16]. This model focuses on leveraging the capabilities of Toffoli gates to enhance the classification process in quantum neural networks. Initially, the input classical data is encoded into quantum states, preparing them for processing within the quantum circuit. The quantum state preparation involves transforming the classical data into quantum form using appropriate encoding schemes. Once the data is encoded, the architecture processes it through several stages of quantum gates and operations.

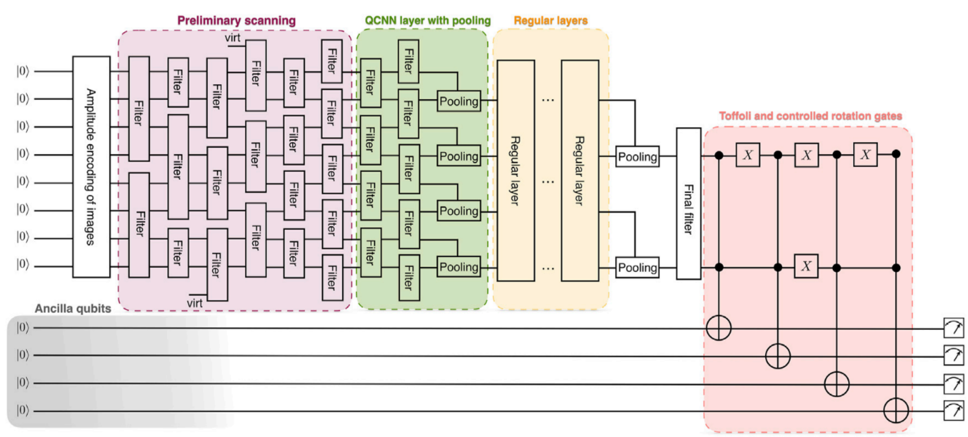


Figure 7: General architecture proposed by Bokhan in [16]

In Figure 7, the global architecture scheme as explicitly detailed in the original paper is presented. The architecture consists of the following key components:

1. **Preliminary Scanning using n-Qubit Filters:** The preliminary scanning layer in our architecture utilizes n-qubit filters to enhance

the flexibility and accuracy of the learning algorithm. Specifically, for 4-qubit filters, the structure involves the application of rotation gates $R_Y(\theta_1)$, $R_Y(\theta_2)$, $R_Y(\theta_3)$, and $R_Y(\theta_4)$ to individually rotate each of the four qubits. This is followed by controlled parameterized rotations $R_Y(\phi)$ for entanglement. The parameterized entanglement scheme is an essential new element in the quantum perceptron structure, providing higher accuracy in image classification by allowing more flexible adjustments to the degree of entanglement. Unlike the standard controlled-X gates used in previous works, the approach with parameterized R_Y gates enables better control over the non-linearities in the learning process, akin to classical activation functions. By transitioning from separable (non-entangled) to entangled states, the quantum system mimics the role of non-linearities in classical learning, thereby improving the overall classification accuracy. This initial scanning with 4-qubit filters sets the stage for subsequent detailed analysis with smaller-scale filters, progressively refining the quantum feature map.

2. **Quantum Convolutional Neural Network Layer with Pooling:** Following the preliminary scanning, the quantum state undergoes further processing with convolutional layers that include pooling mechanisms to distill essential features from the quantum-encoded data. The quantum state of 8 qubits, containing encoded feature maps after the preliminary scanning step, is analyzed in detail by the next layer (see Figure 7). The role of this layer is to extract and prioritize the most significant features from the feature maps. Specifically, in the pooling circuit of Figure 8, a controlled RZ rotation is activated if the first qubit is in state 1, while a controlled RX gate is used when the upper qubit is in state 0. These mechanisms allow the quantum convolutional layers to effectively distill essential information and enhance the discriminative power of the model.

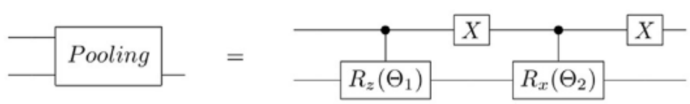


Figure 8: Pooling mechanism using in Bokhan model

3. **Regular Layers:** The architecture incorporates several regular layers after the convolutional layers, enhancing the quantum network's capability to capture detailed patterns and information from the input data. Similar to the classical convolutional neural network (CCNN) approach, our architecture includes 8 additional layers, as depicted in Figure 9. These layers play a crucial role in refining the quantum states further to extract more accurate results. Following individual rotations, double entanglement procedures are introduced to increase the complexity and discriminatory power of the quantum network. This involves implementing entanglement operations between qubits twice within the circuit, akin to the strategy employed in classical CNNs to enhance feature extraction.

   Additionally, after the convolutional layers, two pooling procedures are applied to reduce the number of qubits in the circuit and refine the quantum feature map. The final filter added at the end of the quantum circuit is instrumental in structuring the network to achieve the desired classification outcomes. These architectural enhancements ensure that the quantum network not only processes but also refines and extracts intricate features from the quantum-encoded data, thereby improving its ability to perform accurate and discriminating classifications.

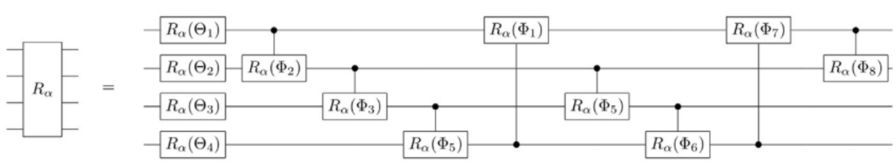


Figure 9: One regular layer used stacked in the final model

4. **Toffoli and Controlled Rotation Gates:** The final layers involve the application of Toffoli gates (controlled-controlled-NOT gates) and other controlled rotation gates. These gates are crucial for performing complex controlled operations necessary for the classification process. The Toffoli gate implementation involves decomposing the gate into two-qubit operations to facilitate practical execution on current quantum hardware, but it's equivalent to the mechanism shown in Figure 10.

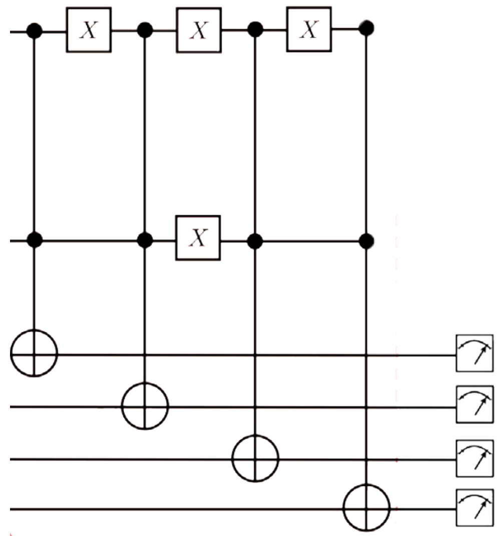


Figure 10: Toffoli gates mechanism proposed by Bokhan in [16]

5. **Measurement:** The quantum state is eventually measured, collapsing it into a classical state. The measurement results correspond to the expected values of each qubit, which are interpreted as the class labels for the input data, concluding the classification process.

By incorporating Toffoli gates, the architecture ensures that the final decision is made through controlled interactions between multiple qubits, enhancing the model's capability to handle complex multiclass classification tasks. This model, grounded in quantum convolutional principles and enriched by the sophisticated logic operations of Toffoli gates, demonstrates robust and accurate classification results, particularly for tasks requiring high precision and the handling of intricate data patterns.

6.3 Description of a Classical Approach: Simple Multilayer Perceptron (MLP)

The classical model used in our study is a simple Multilayer Perceptron (MLP). This model serves as a benchmark to compare the performance of quantum models.

The MLP follows a straightforward data flow, described in figure 11: the input is passed through the first dense layer, followed by activation and dropout, and then through the second dense layer to produce the final output. This architecture allows us to evaluate basic performance and establish a baseline for comparison with more complex quantum models.

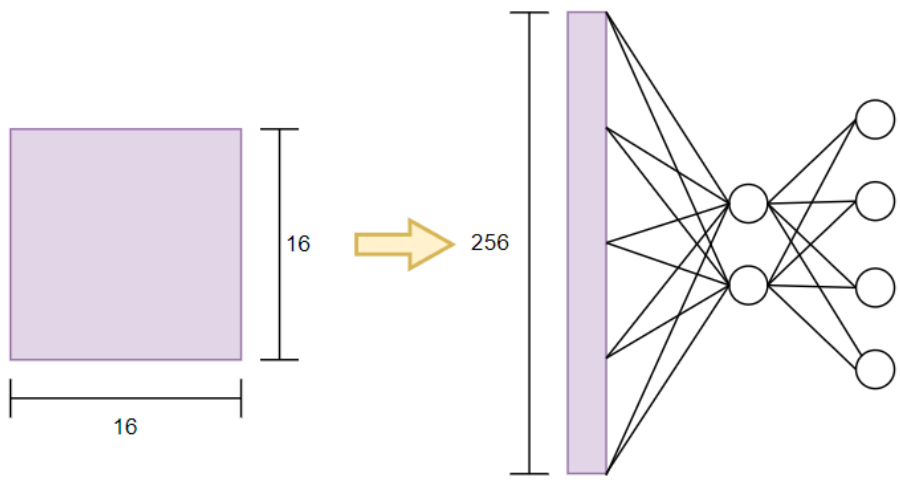


Figure 11: Architecture of the classical model used for comparison. One hidden layer with two neurons and the final layer with four neurons

7 Results

Now, we turn our attention to the presentation of our results. This section will detail the outcomes of our experiments, comparing the performance of quantum and classical machine learning models in the task of classifying spectrogram data. We aim to discuss the relative strengths and weaknesses of each approach, providing a comprehensive understanding of their capabilities and potential applications. All evaluations and metrics have been calculated using the test dataset to ensure an unbiased assessment of model performance.

7.1 Evaluation Metrics

To evaluate the performance of our models, we use three key metrics: accuracy, macro F1 score, and the confusion matrix.

Accuracy is the ratio of correctly predicted instances to the total instances in the dataset. It is a simple and intuitive metric to understand the overall effectiveness of the model. Mathematically, accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{25}$$

This metric provides a direct measure of how often the model's predictions are correct.

We also use the **F1 score** to assess our model. The macro F1 score is the harmonic mean of precision and recall, calcu- lated for each class individually and then averaged. The formula for the F1 score is shown in equation 26. This metric provides a balanced measure of performance across the different classes, taking into account both precision and recall.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{26}$$

Precision is the number of true positive results divided by the number of all positive results, including those not correctly identified. It measures the accuracy of the positive predictions made by the model. The formula for precision is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{27}$$

Recall is the number of true positive results divided by the number of positives that should have been identified. It measures the ability of the model to find all the relevant cases within a dataset. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{28}$$

The **confusion matrix** is a table used to describe the performance of a classification model on a set of test data for which the true values are known. It provides detailed insights into not only the errors being made by a classifier but also the types of errors. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. The diagonal elements represent the number of instances for which the predicted label matches the true label, while off-diagonal elements represent misclassifications as shown in Figure 12.

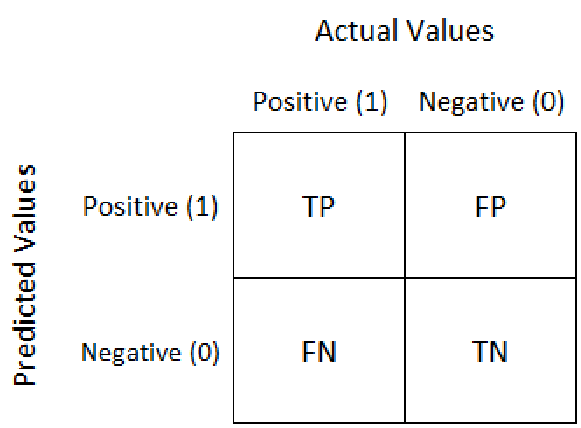


Figure 12: Generic example of the Confusion Matrix

7.2 Cong's Quantum Convolutional Neural Network

The QCNN model proposed in Section 6.1, following the architecture of Cong and incorporating the ancillas from Chamuri, has a total of 204 trainable parameters. To train this model, the following hyperparameters were used: a learning rate of 0.001, a batch size of 8, and 60 epochs. The training process took 43 hours, using Mean Squared Error (MSE) as the loss function.

With these settings, the model achieved an accuracy of **0.692** and a macro F1 score of **0.678**. Although the performance is modest, these results demonstrate that the model is capable of learning certain patterns in the images.

To further illustrate the performance of the QCNN model, the confusion matrix shown in Figure 13 provides a detailed view of the model's classification results. The confusion matrix highlights how well the model classified each class and where it made errors, offering insights into specific areas where the model could be improved.

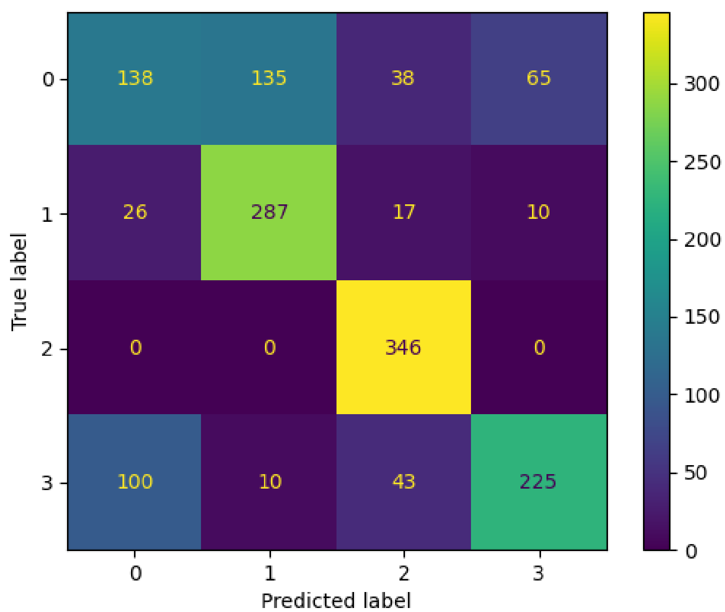


Figure 13: Confusion Matrix of the first quatum model

7.3 Bokhan's Architecture

The QCNN model proposed in Section 6.2, following the architecture by Bokhan that uses Toffoli gates to integrate ancillas, consists of a total of 272 trainable parameters. To train this model, the following hyperparameters were employed: a learning rate of 0.001, a batch size of 32, and 100 epochs. The training process took 45 hours, using Mean Squared Error (MSE) as the loss function.

Under these settings, the model achieved an accuracy of **0.7639** and a macro F1 score of **0.7582**. Although there is still room for improvement, these results surpass the performance of the previous model, demonstrating that the QCNN is capable of learning more complex patterns in the data.

To further illustrate the performance of the QCNN model, the confusion matrix shown in Figure 14 provides a detailed view of the model's classification results. The confusion matrix highlights how well the model classified each class and where it made errors, offering insights into specific areas where the model could be improved.

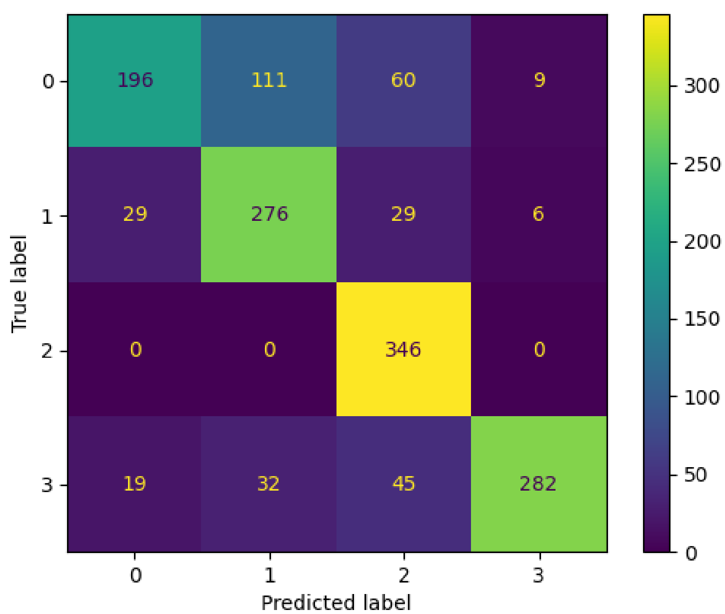


Figure 14: Confusion Matrix of the second quatum model

7.4 MLP Architecture

The MLP model proposed in Section 6.3 consists of a total of 526 trainable parameters. To train this model, the following hyperparameters were employed: a learning rate of 0.0001, a batch size of 64, and 500 epochs. The training process took 2 minutes, using MSE Loss as the loss function.

Under these settings, the model achieved an accuracy of **0.718** and a macro F1 score of **0.696**. These results demonstrate a reasonable level of performance given the simplicity of the model, showing that the MLP is capable of recognizing certain patterns in the images.

To further illustrate the performance of the MLP model, the confusion matrix shown in Figure 15 provides a detailed view of the model's classification results. The confusion matrix highlights how well the model classified each class and where it made errors, offering insights into specific areas where the model could be fine-tuned for even better performance.

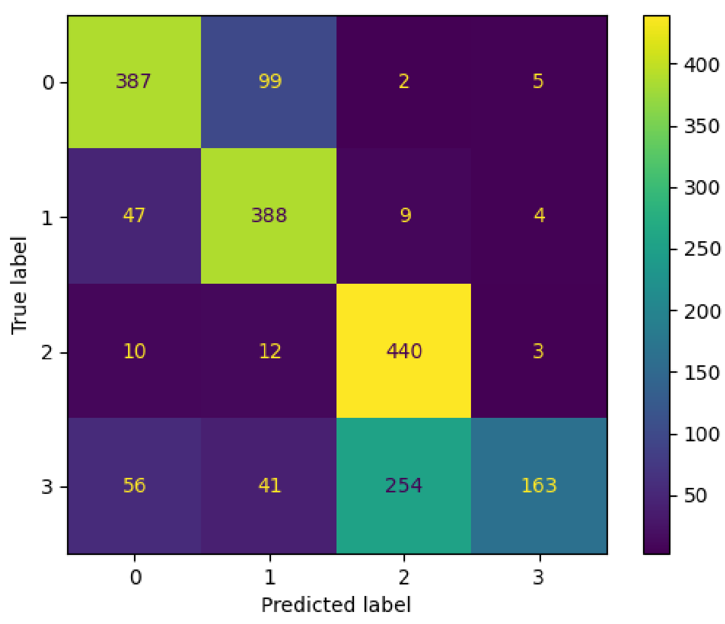


Figure 15: Confusion Matrix of the second quatum model

8 Final Discussion

In Table 1, a summary of results is presented, comparing not only accuracy but also the number of parameters and computational time. While the MLP performed worse than the quantum model in this instance, it is noteworthy that the MLP is designed with simplicity in mind for a fair comparison with quantum models. Although modern convolutional neural networks can achieve accuracies above 0.9 [12], particularly on synthetic datasets, such comparisons would not be equitable with quantum models.

Regarding the performance of quantum models, they demonstrate capability in learning patterns. However, due to the use of simulators, training times are notably prolonged. These algorithms necessitate training in a simulator to infer on quantum computers. Nonetheless, the current state of quantum computers does not yet allow for the implementation of moderately complex models. The precise adjustment of rotational gate angles remains practically unachievable, making it impractical to test models involving nearly 300 rotational gates.

In summary, variational quantum circuits exhibit potential for learning specific patterns. However, the practical application for tasks like image processing or problems requiring classical input-output remains uncertain. These algorithms may find future utility as tools for conducting quantum experiments related to research, where the inputs and outputs are purely quatum.

Model	**Accuracy**	**Nº de Params**	**Comp. Time**
First Quantum Model	0.692	204	43 hours
Second Quantum Model	**0.764**	272	42 hours
Classical Model	0.718	**526**	**4 mins**

Table 1: Accuracy results for the different models used in our study, along with number of parameters and computation time.

9 Conclusion

9.1 Work Review

In conclusion, we can state that the objectives set for this study have been successfully achieved:

Firstly, we designed and implemented a quantum circuit capable of handling image data. This was accomplished by developing a sophisticated quantum architecture that integrates various quantum gates and layers, allowing for the effective encoding and processing of image information.

Secondly, we trained the quantum circuit using a dataset of intrapulse spectrograms. By employing appropriate training algorithms and optimizing hyperparameters, we ensured that the quantum model could learn and adapt to the specific patterns present in the spectrogram data.

Thirdly, we evaluated the performance of the quantum circuit in terms of accuracy and efficiency. The assessment was conducted by measuring key performance metrics, such as accuracy and macro F1 score, as well as considering the computational time required for training.

Finally, we compared the results with classical machine learning approaches to highlight potential advantages and limitations of quantum methods. Despite the simplicity of the classical MLP model used for comparison, the quantum models demonstrated comparable performance, illustrating the potential of quantum computing in pattern recognition tasks. However, the high computational cost and current limitations of quantum hardware were also noted.

Overall, these achievements demonstrate the feasibility and potential of using quantum circuits for image data processing.

9.2 Contributions to the United Nations Sustainable Development Goals

This research contributes to the advancement of several United Nations Sustainable Development Goals (SDGs) by leveraging quantum computing and machine learning techniques to address complex problems. The primary SDGs impacted by this work include:

- **SDG 4: Quality Education**
  The development and dissemination of quantum computing technolo-

gies and methodologies can play a crucial role in enhancing education quality. By integrating advanced topics such as quantum machine learning into academic curricula, we can provide students with cutting-edge knowledge and skills. This fosters a more informed and skilled workforce, capable of driving innovation in various sectors.

- **SDG 9: Industry, Innovation, and Infrastructure**
  Our work on Quanvolutional Neural Networks (QNNs) and their application in classification tasks exemplifies the spirit of innovation and the development of resilient infrastructure. By exploring the potential of quantum computing, we contribute to the creation of new technological paradigms that can lead to more efficient industrial processes and advanced scientific research capabilities.

- **SDG 11: Sustainable Cities and Communities**
  The advancements in quantum computing and machine learning have the potential to transform urban development and management. For instance, improved data processing and analysis can enhance smart city initiatives, optimize resource allocation, and improve urban planning. Our research into sophisticated classification algorithms can be applied to various urban data analytics tasks, contributing to the development of more sustainable and efficient cities.

- **SDG 13: Climate Action**
  By harnessing the computational power of quantum computers, we can better model and understand complex climate systems. This can lead to more accurate climate predictions and effective strategies for mitigating climate change. Our study, which focuses on the practical applications of quantum machine learning, contributes to the broader effort of developing tools and technologies that can address environmental challenges.

- **SDG 17: Partnerships for the Goals**
  The interdisciplinary nature of quantum computing research fosters collaboration across various fields, including computer science, physics, and engineering. By engaging in collaborative research efforts, we support the development of global partnerships aimed at solving complex problems. Our work demonstrates the importance of cross-disciplinary

and international cooperation in advancing technological frontiers and achieving sustainable development.

10 Future Research Lines

The future of quantum computing holds immense promise. As advancements continue, several key areas are expected to shape the development and application of quantum technologies.

Firstly, the future of quantum computers is highly anticipated to be dominated by advancements in photonic quantum computing [19]. Photonic quantum computers use photons as the fundamental unit of information, leveraging the unique properties of light to perform complex computations. These systems offer significant advantages in terms of speed and scalability, as photons can interact with each other in ways that can be harnessed for efficient quantum operations. As research progresses, we can expect photonic quantum computers to play a crucial role in overcoming current limitations related to qubit coherence and error rates.

In the specific context of Quantum Machine Learning (QML), the potential applications are vast and varied. QML can revolutionize fields such as cryptography, optimization, and materials science by providing computational capabilities far beyond those of classical systems. For instance, QML algorithms can be utilized to enhance cryptographic protocols, making them more secure against potential quantum attacks. Similarly, optimization problems that are intractable for classical computers can be tackled more efficiently with quantum approaches, benefiting industries ranging from logistics to finance.

Moreover, QML has the potential to significantly impact the field of artificial intelligence (AI). By leveraging the principles of quantum mechanics, QML algorithms can explore vast solution spaces more effectively, leading to improved performance in tasks such as image recognition, natural language processing, and generative modeling. This convergence of quantum computing and AI could pave the way for the development of highly advanced intelligent systems.

In practical applications, quantum computing is poised to make significant contributions to areas such as drug discovery, climate modeling, and financial modeling. For example, quantum computers can simulate molecular interactions at an unprecedented level of detail, accelerating the discovery of new pharmaceuticals and materials. In climate modeling, the ability to process and analyze vast amounts of data with high precision can lead to more accurate predictions and better strategies for mitigating climate change. In finance, quantum algorithms can optimize trading strategies, risk assessment, and portfolio management, providing a competitive edge in the market.

In the realm of research, quantum computing opens up new frontiers for scientific exploration. Researchers can use quantum systems to simulate complex physical processes, gain deeper insights into quantum mechanics, and develop new theories that were previously unattainable. Additionally, quantum computing can facilitate experiments in quantum physics, enabling the observation and manipulation of quantum phenomena with unprecedented precision.

However, it is important to acknowledge the critical arguments and challenges facing quantum computing. One major concern is the issue of error rates and qubit coherence. Quantum systems are highly susceptible to decoherence and noise, which can lead to errors in computations. Developing robust error correction methods and improving qubit stability are essential for the practical implementation of quantum computers. Additionally, the current limitations of quantum hardware, such as the difficulty in scaling up the number of qubits, the precision required for quantum gate operations and the large times of computing in sumulators, present significant hurdles that need to be addressed.

Overall, the future of quantum computing is incredibly promising, with significant advancements expected in both the technology itself and its wide-ranging applications. As quantum hardware continues to evolve and QML algorithms become more sophisticated, we can anticipate groundbreaking discoveries and innovations that will transform numerous industries and scientific disciplines.

References

[1] Horodecki, R., Horodecki, P., Horodecki, M., & Horodecki, K. (2009). Quantum entanglement. Reviews of modern physics, 81(2), 865.

[2] Rowe, M. A., Kielpinski, D., Meyer, V., Sackett, C. A., Itano, W. M., Monroe, C., & Wineland, D. J. (2001). Experimental violation of a Bell's inequality with efficient detection. Nature, 409(6822), 791-794.

[3] DiVincenzo, D. P. (2000). The physical implementation of quantum computation. Fortschritte der Physik: Progress of Physics, 48(9-11), 771-783.

[4] Glendinning, I. (2005, February). The bloch sphere. In QIA Meeting (pp. 3-18).

[5] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., ... & Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. Reviews of Modern Physics, 94(1), 015004.

[6] Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R., & Van den Nest, M. (2009). Measurement-based quantum computation. Nature Physics, 5(1), 19-26.

[7] Popescu, M. C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems, 8(7), 579-588.

[8] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[9] Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems, 33(12), 6999-7019.

[10] Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. Annals of Data Science, 1-26.

[11] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195-202.

[12] Qu, Z., Mao, X., & Deng, Z. (2018). Radar signal intra-pulse modulation recognition based on convolutional neural network. Ieee Access, 6, 43874-43884.

[13] Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. Nature Physics, 15(12), 1273-1278.

[14] Henderson, M., Shakya, S., Pradhan, S., & Cook, T. (2020). Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. *Quantum Machine Intelligence*, 2(1), 19.

[15] Chalumuri, A., Kune, R., & Manoj, B. S. (2021). A hybrid classical-quantum approach for multi-class classification. Quantum Information Processing, 20(3), 119.

[16] Bokhan, D., Mastiukova, A. S., Boev, A. S., Trubnikov, D. N., & Fedorov, A. K. (2023). Corrigendum: Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. *Frontiers in Physics*, 10, 1112211..

[17] Maouaki, W. E., Marchisio, A., Said, T., Bennai, M., & Shafique, M. (2024). AdvQuNN: A Methodology for Analyzing the Adversarial Robustness of Quanvolutional Neural Networks. arXiv preprint arXiv:2403.05596..

[18] Hur, T., Kim, L., & Park, D. K. (2022). Quantum convolutional neural network for classical data classification. Quantum Machine Intelligence, 4(1), 3.

[19] Zhong, H. S., Wang, H., Deng, Y. H., Chen, M. C., Peng, L. C., Luo, Y. H., ... & Pan, J. W. (2020). Quantum computational advantage using photons. Science, 370(6523), 1460-1463.