Audio Classification Using Quantum Techniques.

by

ARNOB MAJUMDER
24141075

This thesis is submitted to the Department of Computer Science and Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing the degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

_____

Approval

The thesis titled "Audio Classification using Quantum Techniques." submitted by

1. ARNOB MAJUMDER (24141075)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 22, 2024.

Examining Committee:

Supervisor:
(Member)

Shadman Shahriar

Lecturer
Department of Computer Science and Engineering
School of Data and Sciences
BRAC University

Program Coordinator:
(Member)

Dr. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor; Chairperson
Department of Computer Science and Engineering
School of Data and Sciences
Brac University

Acknowledgement

I am highly indebted to my supervisor Mr. Shadman Shahriar, lecturer of the Department of Computer Science and Engineering, BRAC University. This thesis work would never be possible without his guidance and constant supervision of him. While walking on this extremely hard and uncertain path, I only found my supervisor, who held my hand encouraging me to come this far. I am truly grateful to my mentor for his time and effort.

Abstract

Quantum computing is a new type of computing system that is rapidly emerging with immense success in the area of computer science. In our day-to-day lives, there are different types of sounds in our surroundings, which provide us with a lot of information and data. We need to extract the noise and collect important information from it. Convolutional neural networks (CNN) and other techniques have been used for audio classification tasks for several years with high accuracy. But, quantum computing has never been used for audio classifications. So, our goal in this work is to investigate the potential of quantum advantage by experimenting with certain quantum techniques for this specific task. We will scrutinize the effectiveness of the hybrid Quantum Convolutional Neural Network. Also, we check whether it is capable of classifying or optimizing the classification task or not in its Noisy Intermediate Scale-Quantum (NISQ) era.

Keywords: Audio Classification, Quantum CNN, Quantum Techniques

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN Convolutional Neural Network

$MFCC$ Mel Frequency Cepstral Coefficients

$QCNN$ Quantum Convolutional Neural Network

Chapter 1

Introduction

1.1 Motivation

Quantum computing system is a new phenomenon in computer science. In computer science, we traditionally use classical computing systems to calculate and process data. In classical computational systems, we use binary 0 or 1 bit to represent any type of data or information. Along with that, it uses various logic gate which mostly follows the law of classical physics. The classical computational system has come a long way and demonstrates huge success in processing the data accurately. But sometimes it is very hard to calculate or process a large scale of data in this classical computation system. As a result, it gives the wrong output when we want to solve a complex problem. Moreover, it takes a huge time to simulate the behavior of those complex problems. That is the reason for the new computational system, quantum computing. There are such problems where a supercomputer needs a decade to calculate and simulate those problems whereas a quantum computer can easily solve that problem within exponentially the fastest time.

Quantum computing uses the ideas and law of quantum mechanics to calculate the information. In quantum computing, the single bit of expressing the information is known as a qubit. In classical systems, we use just 0 or 1 at the same time but in the quantum system, we can use both 0 and 1 at the same time and all of the possible combinations of that two state by using superposition. That is the main benefit of this computing system because it gives us a multidimensional space to calculate easily and faster which we cannot imagine in classical computing systems. When we pass a piece of information through the quantum computer, it keeps the data in both 0 and 1 bit and uses the probabilistic analysis simultaneously for calculation. When we measure a qubit, then we get the ultimate output. Quantum computing nowadays has a lot of research areas like quantum machine learning, quantum neural networks, quantum cryptography, quantum simulation, quantum algorithms, quantum internet, and so on. Every field gains immense success day by day and also improves for reaching the betterment.

Like classical machine learning, we need quantum machine learning techniques to solve particular problems in a quantum computing system. Quantum machine learning sometimes gives us a broader benefit than classical machine learning. When we use a superposition state, it gives the benefit of calculating in a multidimensional

space. As a result, we can calculate a lot of information in one single time which significantly reduces the total time complexity of any complex problem. Quantum machine learning can be built by using quantum circuits. A Quantum circuit consists of a lot of quantum gates by which we can organize the operation needed to solve that particular problem. To operate the quantum circuit, we need to use some techniques depending on the categories of the problem. Nowadays, many classical machine learning techniques are also implemented in quantum circuits to get extra advantages from them.

In this research work, we take an audio dataset for classification. Nowadays audio or sound can be classified by using various machine learning algorithms which provide highly efficient results. Support Vector Machine, K-nearest neighbor, and many deep learning models like Convolutional Neural Networks are used for achieving the classification task accurately. Our goal in this work is to investigate the potential of quantum advantage by experimenting with certain quantum techniques for the classification and processing of audio. We will test the effectiveness of the Hybrid QCNN and related quantum methods for the audio classification task.

1.2 Problem Statement

In our daily lives, we are surrounded by a lot of sound or audio. We cannot imagine a world without any sound. Sound or audio is a very important material for our life. When we talk to others, we can communicate by voice. People from another side, hear the sound of our voice and respond. When a man coughs, he produces sound, when a man snoring there produces sounds. When the dog barks it creates sounds. When a tiger roars it creates another type of sound which is different from a dog's barking sound. when an aeroplane flies it produces sounds. When it's raining it creates the sound of rain, which is different from a thunderstorm sound. When an alarm bells it produces a sound, when we take water into glass it creates a sound that is different from the wave sounds of the sea. Own creates a sound that is different from the sound of a crow. The guitar produces one type of sound, whereas the violin produces another type of sound. So from morning to night, we engage and experience a lot of sounds. Sound is a very good form of data because it gives us a lot of information. We need to classify which sounds are associated with the things. Audio classification is a very important and significant task in the area of machine learning. Before machine learning, audio classification can be done by frequency and spectral analysis. Using machine learning for the audio classification tasks provides us with very accurate results. We can now correctly predict the sounds of birds, and dogs, snoring of a man, glass breaking, typing on the keyboards, and various sounds by using machine learning algorithms. SVM, KNN, and CNN have already produced very high-accuracy results in this field. Moreover, machine learning researchers research more and more efficient ways to classification of audio data with higher and higher accuracy.
As we know, quantum computing is the future of the new computing era, we explore how the audio classification task can be performed on a quantum computer. To do this, we need to apply various quantum techniques. We scrutinize how quantum techniques can perform on audio classification and is these techniques work well on the classification task for the audio dataset.

1.3 Research Objective

This research aims to explore the effectiveness of various quantum techniques for audio classification tasks. The objectives of this research are:

- To deeply understand how quantum techniques perform in the classification of the audio dataset
- To develop new quantum techniques for the classification problem.
- To save time for classification operations in a short time.
- After completion, we aim to realize our suitable quantum techniques for audio classification and publish our work.

1.4 Research Structure

- In Chapter 1 we discuss the motivation for doing this research including the problem statement and research objective.
- In chapter 2 we discuss some basic definitions of quantum computing. Those basic definitions will help us to understand many terminologies used in our research along basics of quantum information theory.
- In chapter 3 we discuss the literature review.
- In chapter 4, we introduce the dataset analysis where we describe the dataset, some data visualization along data augmentation.
- Research methodology section, we describe in brief how we proceed with our research.
- in chapters 6 and 7 we describe how our model works and its result.
- In chapter 8 we discuss the limitations and advantages of this work.
- Finally, we conclude with chapter 9 and discuss our future plan for this work.

Chapter 2

Quantum Computing Basics

2.1 Qubit

In classical computation, we use a bit where it denotes 0 or 1. In quantum computer systems, it is known as qubits. A two-by-one matrix can be used to represent qubits, which are two-dimensional matrices having complex number elements. It is represented using "Bra-ket" notation. [5] These constitute the computational foundation of qubits.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.1}$$

2.2 Superposition

In addition to 0 or 1, a qubit can remain in various states by using linear state combinations. This characteristic is known as superposition. By using this superposition, we can represent n qubit into a state vector of 2^n Hilbert space. This is the prime advantage that quantum computing provides us but classical computing cannot.

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \quad \text{such that} \quad \|c_0\|^2 + \|c_1\|^2 = 1 \tag{2.2}$$

2.3 Quantum Entanglement

It is a special kind of state that cannot be stated independently. If 2 systems or states are entangled with each other, then, one state is instantaneously effect the other state. Along with that, measuring one state, the other state will also be determined, no matter how much distance there is between those 2 systems. One example of how entanglement states should be:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \tag{2.3}$$

2.4 Measurement

In quantum mechanics, measurement denotes the process of observing the state of a quantum system. After performing the measurement, the superposition state collapses and we get a specific outcome.

2.5 Block Sphere

Any two-level quantum mechanical system's potential states can be represented geometrically by a "Bloch Sphere," Here states are presented as projections that are orthogonal to one another.

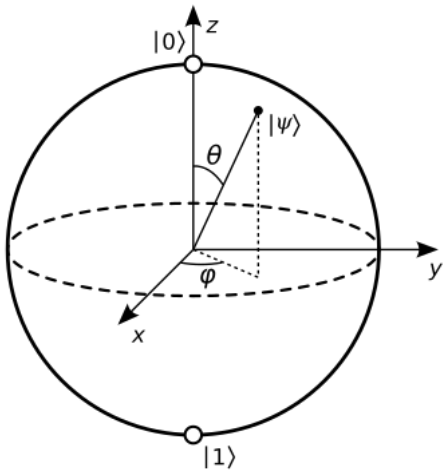

Figure 2.1: Block Sphere

2.6 Quantum Logic Gate

Like classical computing, in quantum computing, we also use many logic gates for performing some necessary operations. Here we discuss some of them that will be further needed in our research. [2]

2.6.1 Pauli Gates

Pauli Gates are quantum logic gates used in various circuits for applying quantum operation. Pauli X gate is known as bit flip operation, Pauli Z is known as phase flip and Pauli y denotes the bit-phase flip operation.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.4}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.5}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.6}$$

2.6.2 CNOT Gate

CNOT gate or Control-NOT gate is one kind of logic gate, where one qubit acts like a control bit and another like a target bit. When the control bit is on 1, then the opposite target will be changed.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.7}$$

2.6.3 Rotational gate

A rotational gate is responsible for rotating on a qubit by the angle in radian with X, Y, and Z axis in a block sphere. [4] Here are the rotational gates used in quantum computation:

$$R_X(\theta) = \begin{bmatrix} \cos\theta & -i\sin\theta \\ -i\sin\theta & \cos\theta \end{bmatrix} \tag{2.8}$$

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.9}$$

$$R_Z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{2.10}$$

Chapter 3

Literature Review

Michael Esposito et al. author a paper [12] where they use quantum machine learning for audio classification with application to healthcare. They implement a hybrid quantum neural network to detect and classify their dataset. They perform it on the COVID-19 cough classification task. After that, they compare classical and quantum neural network methods for this task to see how efficiently classified it can be. From the dataset, they create a Log-mel spectrogram image as feature extraction. The processed image goes through a quantum convolution circuit. Then they pass the image on a classical Recurrent Neural network and a convolutional neural network. The classical RNN gives test accuracy with 79.4% and CNN provides 73% whereas when applying QNN-2 qubits with no noise, it becomes 74.6% and QNN-4 qubits with no noise produce 78.8% accuracy.

Siddhant Dutta, Mann Bhanushali, et al. author a paper [13] about environmental sound classification tasks by using quantum quantized networks. They propose a hybrid QQNN architecture that requires fewer parameters than the normal method. They use the ESC-10 dataset for the task. They use mel spectrogram for feature extraction from the audio dataset pass it on MobileNetV3 architecture and then pass the output into a variational quantum circuit. The training accuracy of classical MobileNet3small was 70.33% whereas the hybrid MobileNetv3small is 85.33%. Moreover, Classical MobileNetv3Large training accuracy was 62.33% but the hybrid one provided 89.67% training accuracy.

The research paper by Joy Krisan Das et al. proposed the idea of using a CNN and LSTM-based system for urban sound classification [11] They used the UrbansSound8k dataset for classification. For feature extraction from the audio dataset, they use MFCC, Mel spectrogram, chroma STFT, and other retable techniques. In the data augmentation part, they use pitch shift, time stretch, and pitch shift with time stretch. From these collected images, they pass these on to the convolutional neural network architecture and observe how correctly it classifies the audio. Along with that, they also pass the signal's images on the LSTM. Applying CNN on the MFCC images they get 90.78% accuracy without augmentation, and 96.78% accuracy with augmentation. Melspectogram images provide output with 83.11 and 94.42% respectively. From LSTM, they got MFCC image accuracy without augmentation 93.30%, with augmentation 98.23%, and mel spectrogram provides 81.85% and 96.25% respectively.

Fan Fan, et al. Authored a paper where they proposed a hybrid quantum-classical CNN model for image classification [14]. In their proposed model, they have used four layers which were encoding layers, quantum convolution layer, measurement layer, and dense layer. When an image passes through the model first it goes through an encoding section. They used the idea of flexible representation of a quantum image for encoding the image [6]. For the quantum convolutional layer, they used a 2*2-sized kernel with a stride size of 2. Then, in the measurement layer, features are mapped into 1D feature vectors for the dense layer. In the dense layer, they implement an activation function to achieve the nonlinear transformation and output as a probability distribution for classifying the category. Their proposed QC-CNN provides 69.7% training accuracy on the overhead-MNIST dataset, 71.8% on the So2Sat LCZ42 dataset, and 85.7% on the PatternNet dataset.

Farina Riaz, et al. propose a neural network model where they use the idea of quantum entanglement approach for image multi-class classification [16]. For implementing this model, they assume that the input image is a 2D matrix of size m*n where the pixel values are normalized. So they use a 4-qubit quantum circuit where 4 pixels are encoded using RY Gate. The output that produces the RY gate is forwarded to the quantum circuit. For this, they use 4 Hadamard gates, 20 three-axis rotations gates, and 20 CNOTS gates. After processing the model, then get output features which are transformed into a 1D vector.

Yijie Dang, et al. use the idea of a quantum K-nearest-neighbor algorithm for the image classification task properly [8]. They compile feature vectors from the images. Then they pass these vectors set on the quantum state for preparation. They calculate the distances between the test images and training images that signify the similarity of they are computed on the quantum circuit and perform amplitude estimation algorithm. To find the k minimum distance from the quantum superposition state they use Durr's Algorithm [3]. Finally, the classification is produced based on the k similarity. They obtain O(root k M). This experiment provides 83.1% accuracy on the Graz-01 dataset and 78% on the Caltech-101 dataset.

Kevin Shen et al. author a paper [17] where they use a variation circuit for data encoding for the classification of the fashion-MNIST dataset. For encoding, they take Flexible Representation of Quantum Images techniques. Then they implement this by introducing the variational algorithms. For a single image, all of the parameters are initialized in the circuit and continuously updated by using a classical optimizer. After initializing the circuit they perform it on 70,000 labeled images for training and perform it on the quantum machine learning techniques.

Debanjan Konar et al. propose random quantum neural networks for recognizing noisy images [15]. They implemented this on the MNIST, fashionMNIST, and KMNIST datasets and obtained an average of 94.9% accuracy. Here they used a classical preprocessing layer for connecting the layer between the temporal pooling layer. After that, the output is encoded as quantum states and passed through a variational quantum circuit.

Irish Cong, et al. proposed a quantum CNN in this paper. [9]. They claim that their QCNN uses O(logN) variational parameters for the input size of N qubits. They construct the QCNN circuit based on two important properties. One is the fixed point criterion. They claim that if the input is a cluster state of L spin, the output of the convolution-pooling layer of one-third of the L. Then another property is the Quantum error correction criterion. According to them, these two properties are necessary for any quantum circuit implementation.

Chapter 4

Dataset Analysis

4.1 Dataset

Dataset analysis is one of the important tasks in the machine learning dataset. Dataset analysis provides a glimpse of the available data of that dataset. We can know the distribution of data, Along with that, we get information about whether there is any error, or null value available in the dataset by performing the data analysis task. That is why it is one of the most important tasks before doing any work in the field of machine learning.

4.2 Data Collection

As we are working on sound classification by using quantum techniques, we use the ESC-50 dataset [7], which is a dataset of environmental sound. This is an open-source dataset. In this dataset, there are 2000 audio data with 50 categories. We preprocess the dataset and use the feature for our further work.

4.3 Dataset Overview

In the ESC-50 open-source environmental sound dataset, there exist 50 categories. Each of the categories contains 40 individual audio data. Dog, glass breaking, sneezing, insects, laughing, washing machine, car horn, clapping, keyboard typing, etc are some of the categories of the dataset. As this is an audio dataset, for feature extraction we need to convert the audio to image by applying some technique to visualize the data. For each audio available in the dataset, we extract Chroma STFT, Mel spectrogram, and MFCC. In total, we have 2000 Chroma STFT images, 2000 Mel spectrogram Images, and 2000 MFCC images.

- **Mel Spectrogram:** It is a combination of spectrogram and Mel scale. Here vertical axis denotes Hz and the horizontal axis denotes time. It provides us the information about the time and frequency. Some of the Mel Spectrogram images are for some existing categories of the dataset.
- **Chroma STFT:** It is another useful feature extraction technique. It maps each STFT bin to chroma after performing a short-time Fourier transform on an audio input. Here vertical axis denotes pitch class and the horizontal

axis denotes time. [11] It provides us the information about the time and frequency. Some of the Chroma STFT images are for some existing categories of the dataset.

- **MFCC:** Mel frequency cepstral coefficient is another technique for audio feature extraction.[1] Here vertical axis denotes frequency cepstral and the horizontal axis denotes time. Some of the MFCC images are for some existing categories of the dataset.

4.4 Data Augmentation

Data Augmentation technique used to increase the existing data. There are many data augmentation techniques for audio datasets like pitch shifting, time stretching, etc. We use the pitch-shifting technique here. We use the factor of +2 and -2 to raise and lower the pitch of the audio.[11] After pitch shifting we will have available 4000 mfcc images of pitch shifting, 4000 mel spectrogram, and 4000 chroma stft images with normal 2000 images of each category.

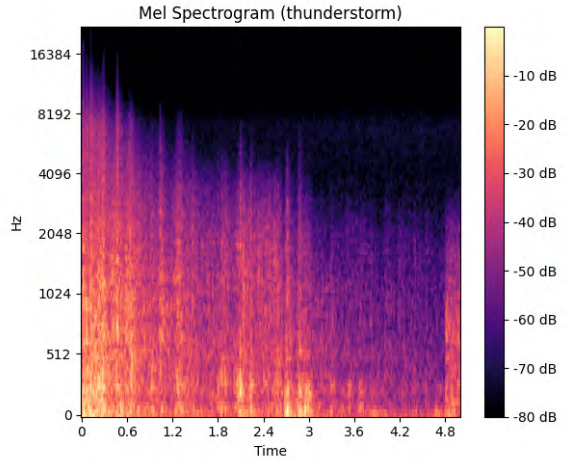

Figure 4.1: Mel Spectrogram of a single Thunderstorm audio

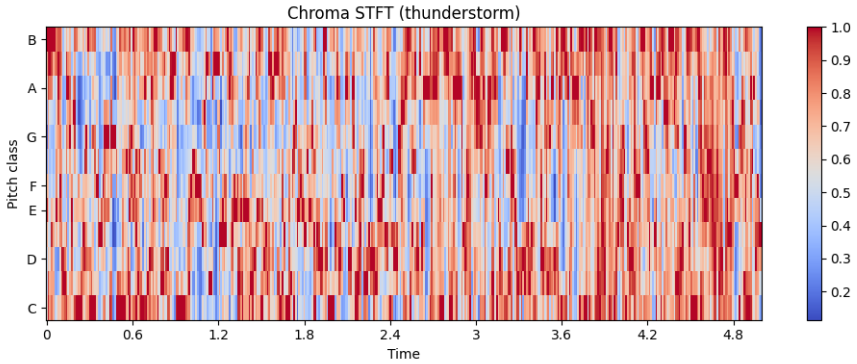

Figure 4.2: Chroma STFT of a single Thunderstorm audio

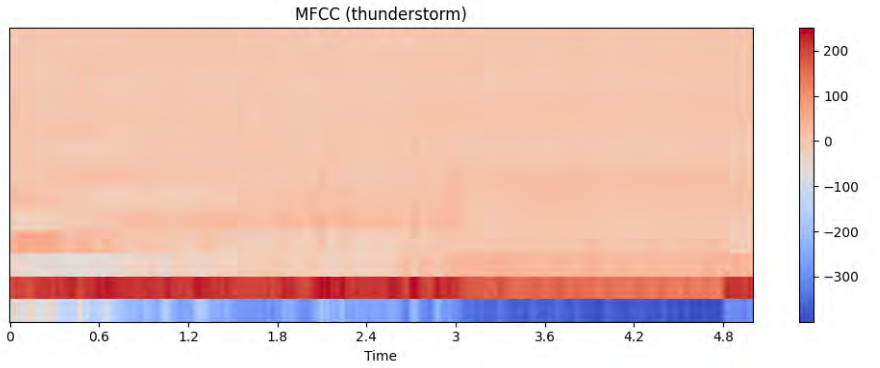

Figure 4.3: MFCC of a single Thunderstorm audio

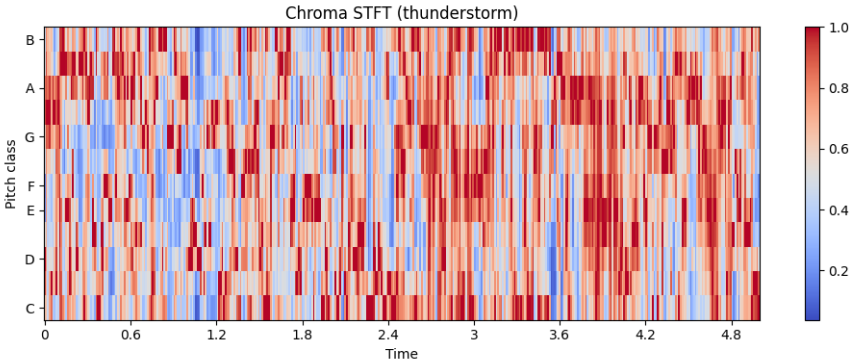

Figure 4.4: Pitch-shift -2 Chroma STFT of a Thunderstorm audio

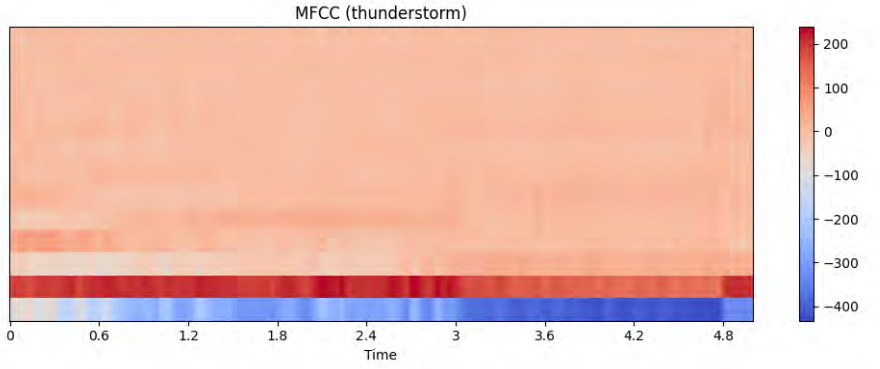

Figure 4.5: Pitch-shift +2 MFCC of a Thunderstorm audio

Chapter 5

Research Methodology

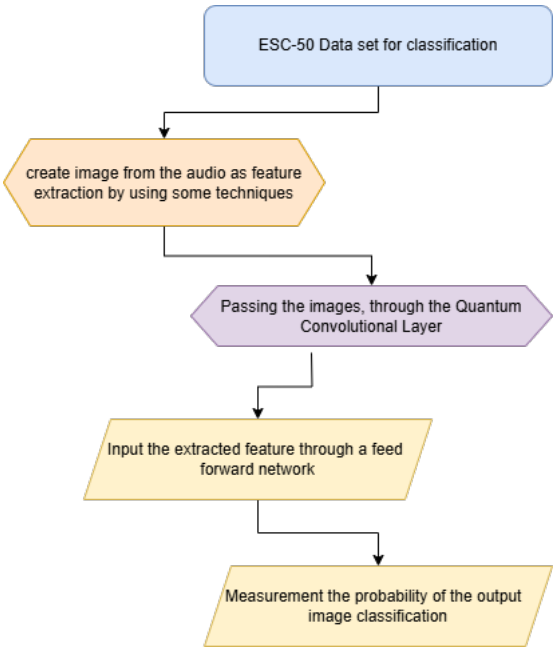

Figure 5.1: Research Methodology

From the dataset, we get 2000 audio. From this 2000 audio, we create 2000 chroma stft, 2000 Mel spectrogram, and 2000 MFCC images. After pitch shifting, we get in total of 18000 images. We perform classical CNN on the images without augmented. On the other hand, we perform QCNN on both augmented and without augmented images. We perform QCNN on augmented data so that we can understand how actually the hybrid model works.
For designing, a quantum circuit for the convolutional layer, we use amplitude embedding for the feature extraction method. A detailed explanation of the model architecture is available in the model architecture chapter.

Chapter 6

Model Architecture

6.1 Convolutional Neural Network

The first model we use here is the convolutional neural network (CNN). CNN is widely used to classify tasks. It is especially well-suited for learning hierarchical feature representations. Generally, CNN extracts spatial features from the input data, increasing the depth of the feature maps while gradually decreasing the spatial dimensions. The architecture starts with several convolutional layers, then moves on to max-pooling layers, and ends with fully linked classification layers. Here is the detail that, I have used in my work:

- **Input Layer**: This layer consists of the size of the images. In this work, we use 400*400*3 where 3 denotes the RGB channels.
- **Convolutional Layers**:
  - 1st layer: this layer consists of 32 filters, each of size 3x3. Here we use the ReLU activation function to introduce non-linearity.
  - 2nd layer: this layer consists of 64 filters of size 3x3. This layer is used to capture more features. Here also, we used ReLU for the activation function
  - 3rd layer: this is the final layer that we used in our work which consists of 128 filters of size 3x3. Here also ReLU activation is applied as well.
- **Max Pooling Layers**: 3 max-pooling layers with a 2x2 window are applied after each of the convolutional layers. This max pooling layer takes the largest value in each 2x2 zone and uses it to minimize the spatial dimensions of the feature maps while keeping the most significant features.
- **Flatten Layer**: the flattening layer used to convert into a 1D vector from the feature maps created by the convolutional layer. After flattening the features, the information is passed to the fully connected layer.
- **Fully Connected Layers**:
  - Dense Layer: A 128-neuron fully connected layer—also called a dense layer—applied on the flattened vector. This layer finds complex patterns from the features retrieved by the convolutional layers. The model is kept

from being a straightforward linear classifier by introducing non-linearity through the use of the ReLU activation function.

– Output Layer: After performing the complex operation with classification, it provides the output. We use the softmax activation function.

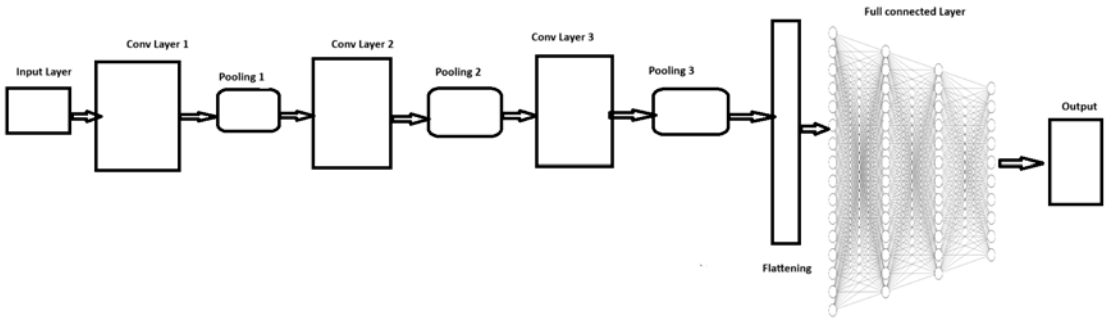

Figure 6.1: CNN Architecture

In the time of training, we use sparse categorical cross entropy and an Adam optimizer. we also implement early stopping to prevent overfitting.

6.2 Quantum Convolutional Neural Network

For our research, we implement a hybrid quantum-classical convolutional network. We design a quantum version of the convolutional layer and pooling layer. Then after applying the quantum convolutional layer and quantum pooling layer on the features, the extracted information will pass on to the classical feed-forward network.

- **image input**: For the lack of computational resources we use here 4 qubits. For this reason, we take 4*4 images so that we can feed the feature perfectly.
- **Quantum Convolutional Layer**:
  - **Amplitude embedding of classical features**: Amplitude embedding is a feature extraction technique that is used to embed classical data into the quantum state. Amplitude embedding's primary benefit is its ability to use the quantum states of very few qubits to represent large amounts of classical data. As we use 4 qubits, So we can extract the $2^4 = 16$ feature by using amplitude embedding. Though we have 4*4*3(RGB) = 48 features, we take only 16 of them. To pass the features, through amplitude embedding, we need to normalize the input features to ensure that, the total probability of all quantum states sum to 1 which is required for amplitude embedding. [10]
  - **Quantum Gates for Entanglement and Rotation**: Quantum gates are applied to the embedded quantum state in this section of the quantum circuit. By adding entanglement and nonlinearity, this technique enables neural networks to learn complex representations. For this, we implement a circuit with CNOT Gate and RX, RY Gate.

* At first, we apply a series of CNOT gates. This gate is used to create correlations between qubits. We use CNOT gates in this layer to entangle adjacent qubits. We connect CNOT between qubit 0 and qubit 1. Here qubit 0 is control and qubit 1 is target. Then qubit 1 and qubit 2. In this case, qubit 1 acts like control, and qubit 2 becomes the target. and finally qubit 2 with qubit 3. These gates enable the network to identify patterns in the incoming data and record interactions between adjacent qubits.
* Rotation gates are applied around the X-axis (RX) and Y-axis (RY) after the CNOT gates. Learnable parameters are added to the network through the rotation gates. These gates use an angle corresponding to the parameter (0.2 * i for RX and 0.3 * i for RY) to rotate the state of each qubit. For example, in qubit 2, it is, RX(0.2*2) = RX(0.40) and RY(0.3*2)= RY(0.60). These parameters regulate how much rotation occurs and introduce nonlinearity into the quantum circuit.

– **Measurement**: The measurement of the qubits is the last stage of the quantum layer. Here, we quantify each qubit in Pauli-Z expectation value. We can determine the probability along the Z-axis. The expected value, which ranges from -1 to 1, is returned by this measurement. Suppose, the outcome is |0> so the measurement is +1 which is known as spin up and for |1> the measurement is -1 known as spin down. After that, the value will be passed to the next layer for further processing.

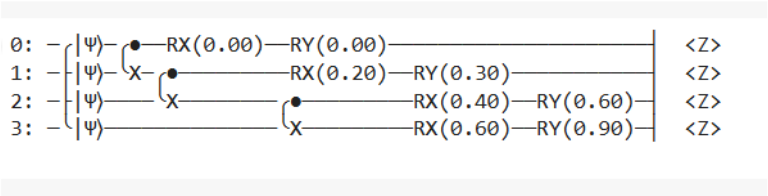

Figure 6.2: Quantum circuit of Quantum convolutional Layer

- **Quantum Pooling Layer**: In classical CNN, there is a pooling layer after a convolutional layer. Similarly, we introduce a mechanism for quantum computation. It is responsible for reducing the number of qubits and reducing the spatial dimension. The features that came from the convolutional layer have become the input feature for this quantum pooling layer. After that, we use RX and RY rotation. The RX applies in the x-axis of the block sphere for the qubit at the I index. Same as RY corresponding to the y-axis. How much it will rotate, depends on the value of input features it gets. After quantum rotations to all qubits, we perform a pooling operation to reduce the number of qubits and the information of that subsequent layer. In this case, we reduce the number of qubits from 4 to 2. As a result, the model extracts key information about the quantum system and finally measures it in Pauli-Z expectations. By this measurement, we only get classical information based

on the quantum states of those qubits. After that, the output is fed into the fully connected layers.

```
0: ──RX(-0.64)──RY(-0.64)─┤  <Z>
1: ──RX(-0.94)──RY(-0.94)─┤  <Z>
2: ──RX(-0.05)──RY(-0.05)─┤
3: ──RX(-0.09)──RY(-0.09)─┤
```

Figure 6.3: Quantum circuit of Quantum pooling Layer

- **Classical Flattening and fully connected Layer**: After performing this quantum operation, now we pass the information that we get from the quantum state to the flattening layer which is used to convert it into a 1D vector from the feature maps. After flattening the features, the information is passed to the fully connected layer. On the flattened vector, a 64-neuron fully connected layer is applied. This layer finds complex patterns from the features retrieved by the quantum convolutional layers. The model is kept from being a straightforward linear classifier by introducing non-linearity through the use of the ReLU activation function.

While training the model, We ensured the setup was just like the classical CNN. Below is the basic architecture of our hybrid quantum CNN.

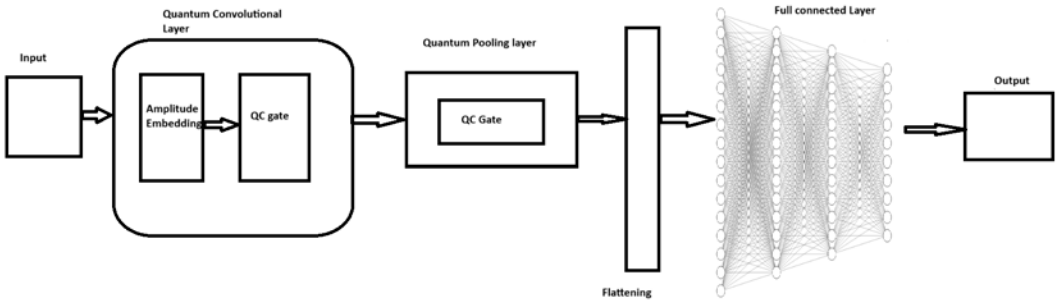

Figure 6.4: QCCN Architecture

Chapter 7

Result Analysis

7.1 Test Accuracy

7.1.1 Test accuracy of classical CNN

In the current setup where the dense layer is set to 128, after performing a classical convolutional neural network we achieved 98% accuracy in the Mel Spectrogram category, 100% in the MFCC category, and 98% in the Chroma STFT image category. All of these 3 categories have individually 2000 images without any augmentation. We split the dataset into 1280 training samples, 320 validation samples, and 400 test samples.

Image Category	Epoch	Batch Size	Image Size	Test Accuracy
Mel Spectrogram	10	16	400*400	98%
MFCC	10	32	400*400	100%
Chroma STFT	10	32	400*400	98%

Table 7.1: Performance of Classical CNN

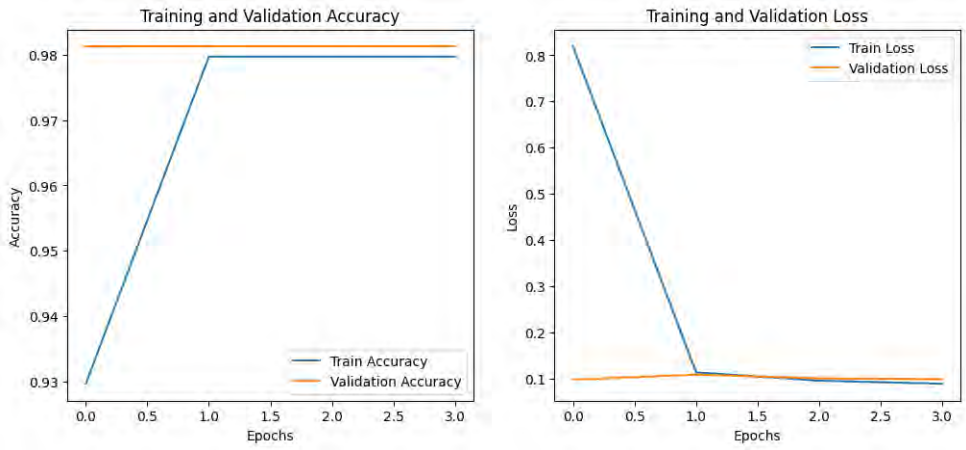

Figure 7.1: CNN performance on Chroma STFT

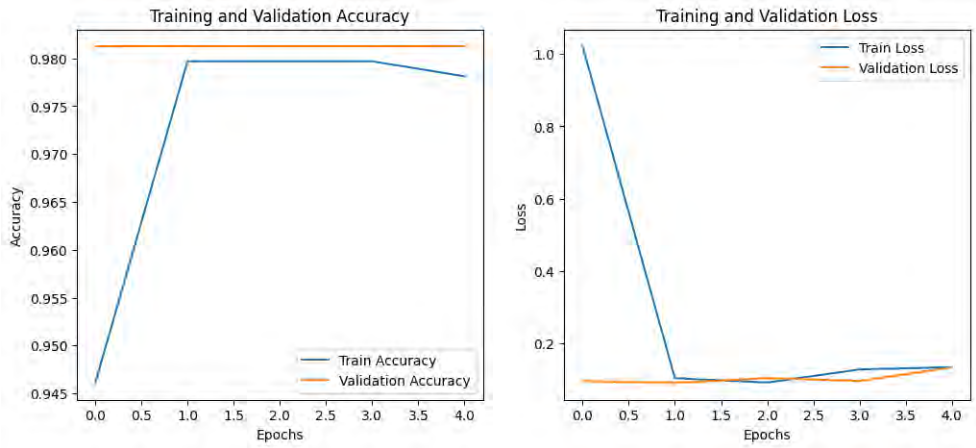

Figure 7.2: CNN performance on Mel Spectrogram

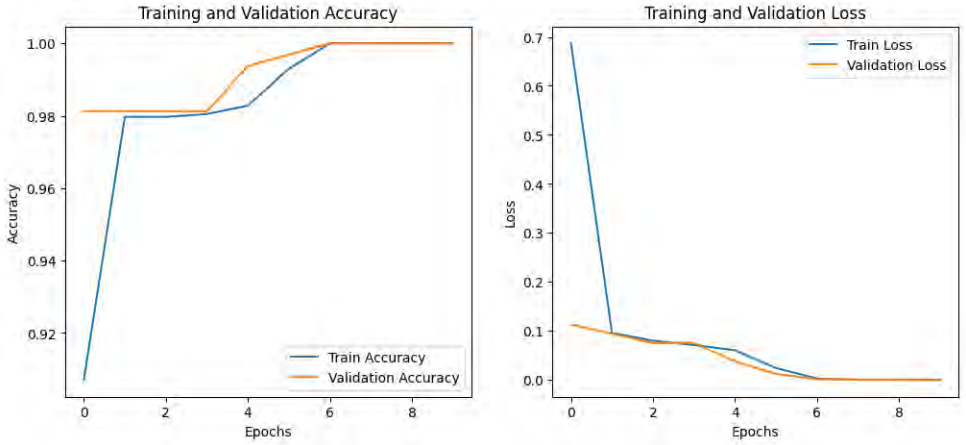

Figure 7.3: CNN performance on MFCC

These accuracies on image categories were achieved with high accuracy as the number of images per category is not so many. Normally, it is predicted that CNN provides a very very high accuracy with this dataset as it takes only 1280 training samples and gives tests on just 400 images. After extracting the feature from the images by 3 convolutional layers and 3 pooling layers and finally feeding it into a 128-dense layer, it is very obvious that the classical CNN performs very well in this classification task.

- **Mel Spectrogram**: During the training period of the Mel spectrogram, we see that the training accuracy increases rapidly during the first epoch which indicates the model is learning very well from the training dataset. The validation accuracy is also 98% and remains constant throughout the training process which signifies that the accuracy of the validation data may not improve over time. But in training loss and validation loss is very low which we understand that the performance of the model is good and does not overfit significantly.
- **MFCC**: We see that the training accuracy from 0 to 4 epochs reaches close to 100%. Validation accuracy is also stable at around 98%. On the other hand, if we observe the training and validation loss, we see that they are converging

toward a very low value which means the model might not overfitting.

- **Chroma STFT**: Here also the training and validation accuracy is 98% and loss is very low. This helps us to understand that, in this category also the cnn performs very well. We also see that, Along with that, the difference between training and validation loss is very low.

7.1.2 Test accuracy of Quantum CNN

In the current setup after performing a hybrid quantum convolutional neural network we achieved 98% accuracy in all of the image categories. All of these 3 categories have individually 2000 images without any augmentation where we split the dataset into 1280 training samples, 320 validation samples, and 400 test samples. Along with that, to ensure the model works perfectly on the large dataset, we use augmented data also. After using Augmented data in 3 categories we have 6000 images per category. We split the dataset into 3840 training samples, 960 validation samples, and 1200 test samples.

Image Category	**Epoch**	**Batch Size**	**Image Size**	**Accuracy with Augmentation**
Mel Spectrogram	20	8	4*4	98%
MFCC	20	8	4*4	98%
Chroma STFT	20	8	4*4	98%

Table 7.2: QCNN Test accuracy with augmented image

Image Category	**Epoch**	**Batch Size**	**Image Size**	**Accuracy with Augmentation**
Mel Spectrogram	20	8	4*4	98%
MFCC	20	8	4*4	98%
Chroma STFT	20	8	4*4	98%

Table 7.3: QCNN Test accuracy without augmented image

Though we achieved 98% accuracy during the training, there is something that needs to be observed. We use 4*4 images. We are unable to feed high pixels because of the lack of the qubit. As we discussed earlier, we use just 4 qubits in our research so, 4 qubits can process only 16 features. So the number of features fed into the model is very low. When we fed the 16 features into the quantum convolutional layer, the layer provided output with just 4 features. These 4 features went through a pooling layer and reduced 2 more features. So finally it processes only 2 features when it goes to the 64 dense layer. As a result, during training, the model did not find any difficulties. So we achieved very high accuracy. Here as the feature number is low it was possible to provide the low accuracy but it did not.

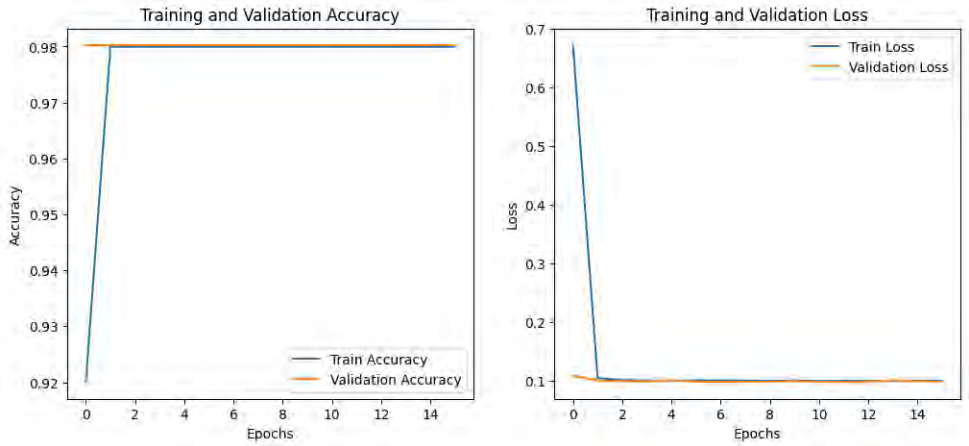

Figure 7.4: QCNN performance on Augmented Chroma STFT

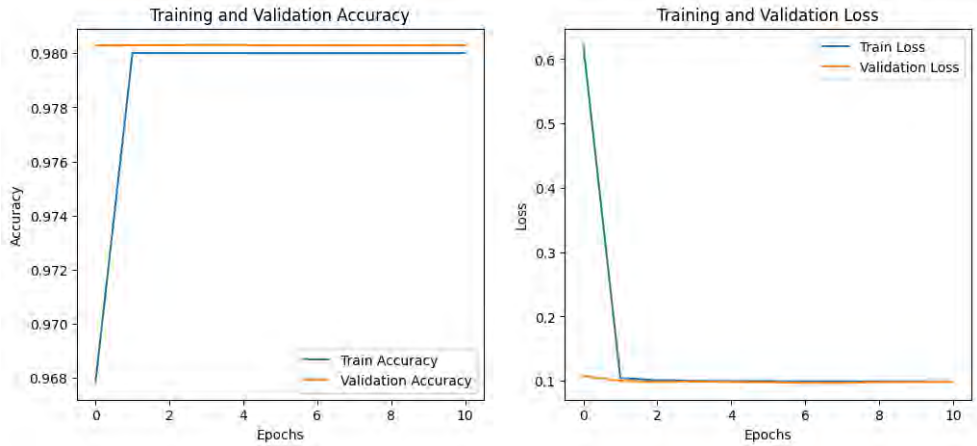

Figure 7.5: QCNN performance on Augmented MFCC

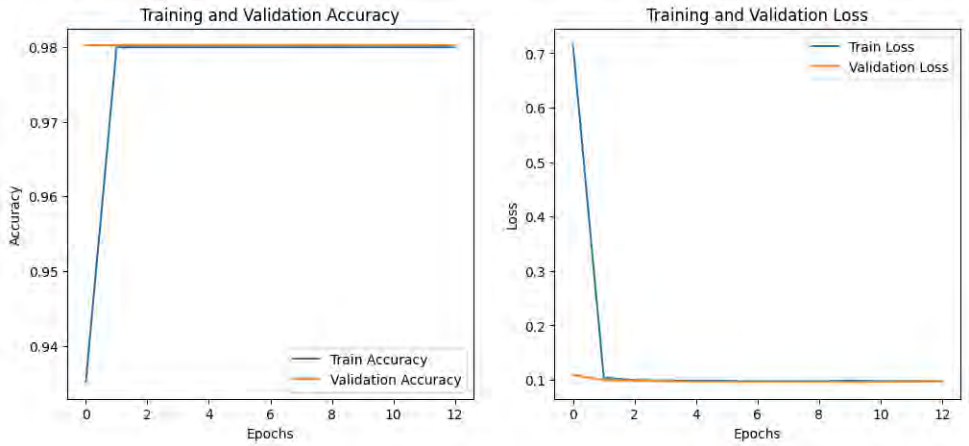

Figure 7.6: QCNN performance on Augmented Mel Spectrogram

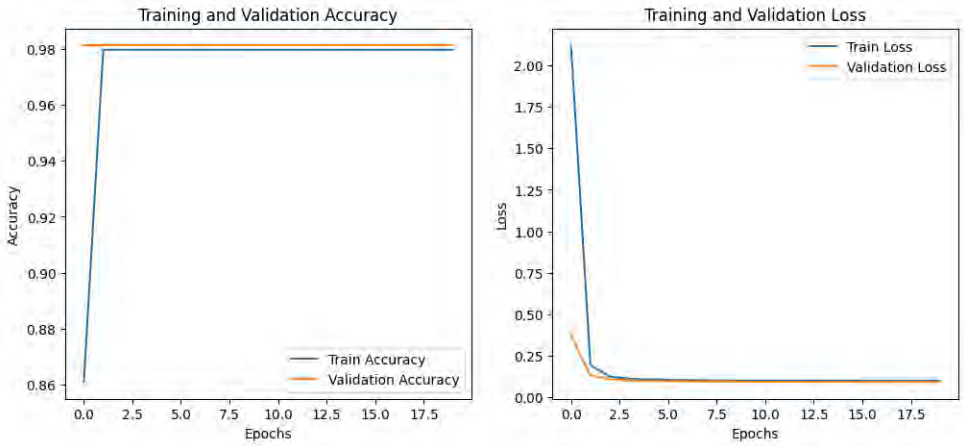

Figure 7.7: QCNN performance without Augmented Chroma STFT

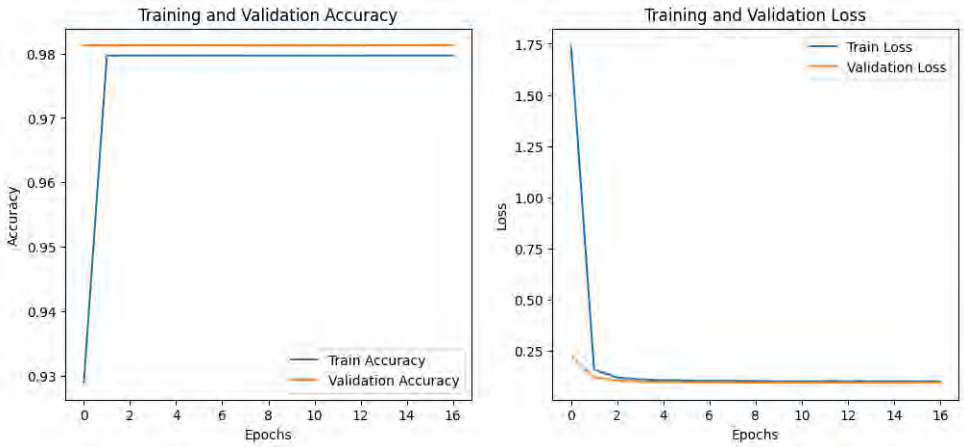

Figure 7.8: QCNN performance without Augmented MFCC

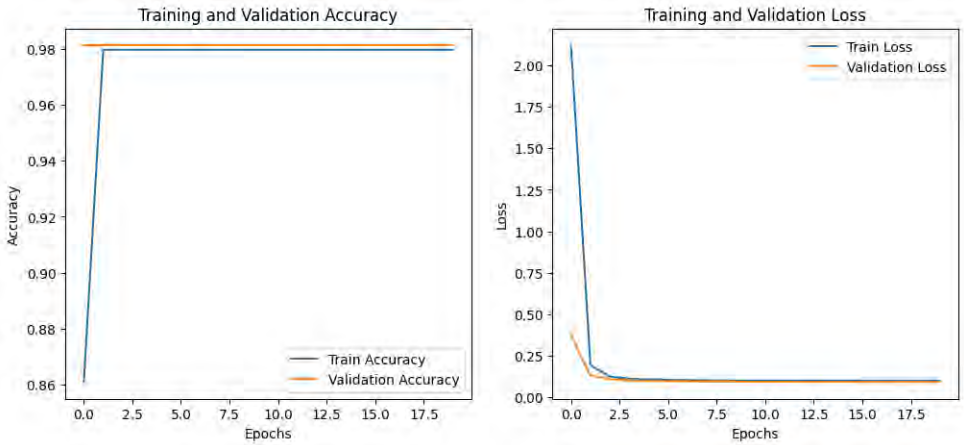

Figure 7.9: QCNN performance without Augmented Mel Spectrogram

7.2 Trainable Parameter

7.2.1 Classical CNN

When performing classical CNN on our image dataset, for 400*400 images and setting the dense layer to 128, we get 37,848,562 trainable parameters in total which is approximately 37 Million! Below is the breakdown of the parameters:

Layer (type)	**Parameter #**
conv layer 1	896
Pool layer 1	0
conv layer 2	18,496
Pool layer 2	0
conv layer 3	73,856
Pool layer 3	0
flatten layer	0
dense	37,748,864
dense_1	6,450
Total parameters	**37,848,562**
Trainable parameters	**37,848,562**
Non-trainable parameters	**0**

Table 7.4: CNN Model summary with parameters

Moreover, if we use 128*128 size images, and a dense layer with 128 we get trainable parameters based on the CNN model is 3,311,090 in total which is approximately 3.3 Million! Below is the breakdown of the parameters based on the dense layer:

Input Image Size	**Dense Layer**	**Trainable Parameter**
128*128	64	1,702,194
1024*1024	64	130,152,754
1024*1024	128	260,212,210

Table 7.5: Trainable parameters for different input image sizes and dense layers.

It is happening because, in classical CNNs, the network learns the parameters of each filter during training. So when we input large pixels of images, adding more filters and layers quickly increases the number of parameters.

7.2.2 Quantum CNN

Now we calculate how many parameters are used here in our designed hybrid quantum convolutional neural network:

- Quantum Convolutional Layer:
  - We use 4 qubits, for processing 16 features.
  - We use 2 types of Rotational Gate. So, for 4 qubits * 2 parameters per gate = 8 parameters.
- Quantum Pooling Layer:
  - In the pooling layer, we reduce qubits from 4 to 2. Each qubit in this layer also has 2 trainable rotational gates.
  - So, We use 2 types of Rotational Gate. So, for 2 qubits * 2 parameters per gate = 4 parameters.
- Flatten Layer: It is used for reshaping the data.
- Dense Layer:
  - After the pooling layer we get the output of a vector of size 2.
  - it takes 2 features as input and output 64 units.
  - So, 64*2 +64 (bias) = 192 Parameters
- Output Layer:
  - There are 50 available classes in my dataset.
  - So, 50*64 + 50(Bias) = 3250 Parameters.

  Total Parameters = Quantum Conv. Layer+ Quantum Pooling Layer+ Dense Layer + Output layer = 8 + 4 + 192 + 3,250 = 3,454 trainable parameters.

  Now, this parameter calculation is based on 4 qubits which process initially 4*4 size image means 16 features. So, mathematically, if the number of qubits can be increased, in that case, we can feed more features. In that case, the number of trainable parameters will be(Assuming 64 dense layers):

Image Size	**Required Qubit**	**Total Parameter**
8*8	6	3590
16*16	8	3726
32*32	10	3862
128*128	14	4134
1024*1024	20	4542

Table 7.6: total parameters for different image sizes in QCNN

7.3 Observation

- **Based on Test Accuracy**:
  - Though we get about 98% accuracy in both case but:
    - As the image size fed into the hybrid quantum convolutional neural network during training is very low, it does not face any difficulty. As a result, it provides high accuracy based on the small features it gets.
    - the accuracy that we get from the hybrid QCNN, that does not beat the result of classical CNN. So, we cannot conclude that, the hybrid quantum CNN performs better than the classical one. Machine learning nowadays has so many advantages that, it is nearly impossible to beat the result for comparatively new technology.
- **Based on the trainable parameter**:
  - Here we see that, quantum computing, allows us to use more features by using fewer parameters. It is happening because of the advantage of superposition. We can input 2^n features by using just n qubits. that's never possible for any classical computing system. So, based on the parameter, QCNN beat the classical CNN.

Chapter 8

Limitations and Advantage

8.1 Limitations

- The main limitation of quantum computing is the lack of computational resources. As it is still in the noisy intermediate scale quantum(NISQ) era, it is difficult to use more qubits(quantum units) right now.
- We use a hybrid model here. This means the convolutional layer and the pooling layer that are made off with the quantum, after that, we calculate the extracted feature classically and feed it to the classical neural network.
- Quantum Computing is in its early stage. Same as, the classical computer in the 70s or 80s era. So it is very tough for quantum computing to beat the accuracy of Machine learning right now. Classical Machine learning performs so well that, it almost provides us with very high accuracy in many classification tasks.
- This qubit is not inserted in classical computers right now. We have to use some large company's servers for simulations. When it will be available on every device, it will create immense success.

8.2 Advantage

- Superposition is the main advantage. If the number of qubits can be increased, lots of work can be done so easily. Suppose someone has just 100 qubits, then he can process 2^{100} information!
- For feature extraction we use quantum amplitude encoding. This technique is more helpful for extracting the information than any other classical tool. Moreover, we use rotational gates where we used to rotate the qubit in radians based on the x and y axis. As a result, there are more complex operations happening inside it for feature extraction. Which are better than using a single non-linear function in the layer.
- We see that it allows us to use fewer parameters than the classical one. Fewer parameter signifies that it takes less memory and less time for training! So, in very large-scale work suppose in various big-data related works it provides the advantage.

Chapter 9

Future Work and Conclusion

9.1 Future Work

I am planning to add the number of quantum layers into the quantum convolutional layer and pooling layer to see how it will behave on the complex networks. Moreover, I will implement the effective quantum version of the classical Support vector machine(SVM) and K-nearest neighbor algorithm(KNN) for audio classification in the future.

9.2 Conclusion

In this paper, we talked about the audio classification task using quantum techniques. We propose an approach to implement a quantum convolutional neural network for this audio classification task. Though Audio classification can be done by classical machine learning algorithms with high accuracy, we want to explore how it will perform in the quantum computer. For implementing this on the quantum computer, we measure the effectiveness of some techniques that are fruitful for classifying this task in quantum computers. Along with that, we can conclude that the number of trainable parameters is less than the classical Convolutional neural network. It signifies the advantage of the quantum era in the upcoming time.

Bibliography

[1] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980. DOI: 10.1109/TASSP.1980.1163420.

[2] A. Barenco, C. H. Bennett, R. Cleve, *et al.*, "Elementary gates for quantum computation," *Physical Review A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995, ISSN: 1094-1622. DOI: 10.1103/physreva.52.3457. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.52.3457.

[3] C. Durr and P. Hoyer, *A quantum algorithm for finding the minimum*, 1999. arXiv: quant-ph/9607014 [quant-ph].

[4] N. S. Yanofsky and M. A. Mannucci, *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[6] K. H. Phuc Q. Le Fangyan Dong, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," *Quantum Information Processing*, vol. 10, pp. 63–84, 2011. DOI: 10.1007/s11128-010-0177-y.

[7] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia: ACM Press, Oct. 13, 2015, pp. 1015–1018, ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806390. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[8] Y. Dang, N. Jiang, H. Hu, Z. Ji, and W. Zhang, *Image classification based on quantum knn algorithm*, 2018. arXiv: 1805.06260 [cs.CV].

[9] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[10] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical Review Letters*, vol. 122, no. 4, Feb. 2019, ISSN: 1079-7114. DOI: 10.1103/physrevlett.122.040504. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.122.040504.

[11] J. K. Das, A. Ghosh, A. K. Pal, S. Dutta, and A. Chakrabarty, "Urban sound classification using convolutional neural network and long short term memory based on multiple features," in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, 2020, pp. 1–9. DOI: 10.1109/ICDS50568.2020.9268723.

[12] M. Esposito, G. Uehara, and A. Spanias, "Quantum machine learning for audio classification with applications to healthcare," in *2022 13th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2022, pp. 1–4. DOI: 10.1109/IISA56318.2022.9904377.

[13] S. Dutta, M. Bhanushali, S. Bhan, L. Varma, P. Kanani, and M. Narvekar, "Quesc: Environmental sound classification using quantum quantized networks," *Procedia Computer Science*, vol. 230, pp. 554–563, 2023, 3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks (ICECMSN 2023), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2023.12.111. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923021166.

[14] F. Fan, Y. Shi, T. Guggemos, and X. X. Zhu, "Hybrid quantum-classical convolutional neural network model for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023. DOI: 10.1109/TNNLS.2023.3312170.

[15] D. Konar, E. Gelenbe, S. Bhandary, A. Das Sarma, and A. Cangi, "Random quantum neural networks for noisy image recognition," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 02, 2023, pp. 276–277. DOI: 10.1109/QCE57702.2023.10240.

[16] F. Riaz, S. Abdulla, H. Suzuki, S. Ganguly, R. C. Deo, and S. Hopkins, "Accurate image multi-class classification neural network model with quantum entanglement approach," *Sensors*, vol. 23, no. 5, p. 2753, 2023.

[17] K. Shen, B. Jobst, E. Shishenina, and F. Pollmann, *Classification of the fashion-mnist dataset on a quantum computer*, 2024. arXiv: 2403.02405 [quant-ph].