

**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
PULCHOWK CAMPUS**

**THESIS NO: 076MSDSA002  
HYBRID QUANTUM-CLASSICAL DEEP LEARNING MODEL FOR  
PREDICTION OF COVID-19 USING CHEST X-RAY IMAGES**

**BY  
ANISH BARAL**

**A THESIS REPORT  
SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND  
COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF MASTER IN COMPUTER  
ENGINEERING, SPECIALIZATION IN DATA SCIENCE AND  
ANALYTICS**

**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
LALITPUR, NEPAL**

**SEPTEMBER, 2022**

**HYBRID QUANTUM-CLASSICAL DEEP LEARNING MODEL FOR  
PREDICTION OF COVID-19 USING CHEST X-RAY IMAGES**

BY

ANISH BARAL

076/MSDSA/002

Thesis Supervisor

Prof. Dr. Shashidhar Ram Joshi

A thesis submitted in partial fulfillment of the requirements for the degree of  
Master in Computer Engineering, Specialization in Data Science and Analytics  
(MSDSA)

Department of Electronics and Computer Engineering  
Institute of Engineering, Pulchowk Campus  
Tribhuvan University  
Lalitpur, Nepal

SEPTEMBER, 2022

## COPYRIGHT©

Author, thesis supervisor and Institute of Engineering, Pulchowk Engineering Campus, Department of Institute of Engineering, Pulchowk Engineering Campus, has reserved all right on this dissertation for its distribution and for scholarly purpose. In the absence of author and supervisor one can take permission from Department of Electronics and Computer Engineering. Without the prior permission of the author, supervisor or Department of Electronics and Computer Engineering this publication should not be reproduced, transmitted through any mechanical or electronics medium and should not be distribution through the photocopy. For research and academic purpose the dissertation can be freely used under the permission. Distribution of this publication for the financial gain of for the personal benefit is strictly prohibited.

Request for permission to copy or to make any use of the material in this thesis in whole or part should be addressed to:

Head

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Pulchowk, Lalitpur, Nepal

## DECLARATION

The dissertation is submitted for fulfilment of degree in **Master in Computer Engineering, Specialization in Data Science and Analytics** to Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, under the title “**HYBRID QUANTUM-CLASSICAL DEEP LEARNING MODEL FOR PREDICTION OF COVID-19 USING CHEST X-RAY IMAGES**”. I swear this work has not been done previously and is completely my own work. This work has not been previously submitted to university to fulfil any academic award.

Department of Electronics and Computer Engineering, Pulchowk Campus has authorized to freely distribute this thesis to other intuitions or to any individuals for further scholarly research work.

Anish Baral

076/MSDSA/002

Date: SEPTEMBER, 2022

## RECOMMENDATION

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled **“HYBRID QUANTUM-CLASSICAL DEEP LEARNING MODEL FOR PREDICTION OF COVID-19 USING CHEST X-RAY IMAGES”**, submitted by **Anish Baral** in partial fulfillment of the requirement for the award of the degree of **“Master in Computer Engineering, Specialization in Data Science and Analytics”**.

.....

**Supervisor: Professor Dr. Shashidhar Ram Joshi,**  
**Department of Electronics and Computer Engineering,**  
**Institute of Engineering, Tribhuvan University**

.....

**External Examiner:**

.....

**Committee Chairperson: Asso. Professor Dr. Arun Kumar Timalina**  
**Program Coordinator MSDSA,**  
**Department of Electronics and Computer Engineering,**  
**Institute of Engineering, Tribhuvan University**

## DEPARTMENTAL ACCEPTANCE

The thesis entitled “**HYBRID QUANTUM-CLASSICAL DEEP LEARNING MODEL FOR PREDICTION OF COVID-19 USING CHEST X-RAY IMAGES**”, submitted by **Anish Baral** in partial fulfillment of the requirement for the award of the degree of “**Master in Computer Engineering, Specialization in Data Science and Analytics**” has been accepted as a bonafide record of work independently carried out by him in the department.

.....

**Prof. Dr. Ram Krishna Maharjan**

Head of the Department

Department of Electronics and Computer Engineering,

Pulchowk Campus,

Institute of Engineering,

Tribhuvan University,

Nepal.

## ACKNOWLEDGEMENT

I would like to thank the **Department of Electronics and Computer Engineering, IOE, Pulchowk Campus** for giving me the opportunity to work on this thesis as part of the Masters of Science in Data Science and Analytics (MSDSA) syllabus.

My deep gratitude goes to my supervisor **Prof. Dr. Shashidhar Ram Joshi** for his patient guidance, eager encouragement and valuable critiques on this thesis work. Equally I would express my very great appreciation to my program coordinator **Dr. Arun Kumar Timalsina** for his continuous support, valuable and constructive suggestions. His willingness to give his time so generously has been very much appreciated.

I am very grateful and pay my sincere gratitude to all the teachers of Department of Electronics and Computer Engineering of **Institute of Engineering, Pulchowk Campus** for providing opportunity of learning and implementing the knowledge in the form of research work.

Any other suggestions or criticisms of the improvement of this work will be greatly appreciated.

Anish Baral

076/MSDSA/002

## ABSTRACT

Medical images are difficult to collect and are full of insecurities and expensive-ness. Pandemic such as COVID-19 break out suddenly and may be transferable from one person to another ,so we need to identify the victim and isolate them. Presence of less datasets of such cases are difficult for the classical convolution model for prediction of disease. We need a high performance and accurate image classification model that assists doctor in diagnosis. The CNN layer of deep learning is also computationally complex as it need a lot of weights to train for better performance ,this increases the computational complexity of the model. Therefore, it is very necessary to develop a model which is fast, accurate and computationally efficient model. Here, we present a hybrid quantum classical convolution neural network for image classification. We run the model in simulators and different real quantum devices. We found that the hybrid model with less trainable parameters with low resolution and small training images was able to outperform the classical convolution neural network. The best hybrid quantum -classical model in this work was with accuracy of 0.9348 and 12318 trainable parameters. The best classical model was with accuracy of 0.9076. The computationally efficient model was with accuracy of 0.9239 with 2355 learn able parameters.

**Keywords:** Quantum Computing, Quantum Convolution Neural Network, COVID-19, Hybrid Quantum-Classical Model



## Contents

<b>COPYRIGHT</b>	<b>iii</b>
<b>DECLARATION</b>	<b>iv</b>
<b>RECOMMENDATION</b>	<b>v</b>
<b>DEPARTMENTAL ACCEPTANCE</b>	<b>vi</b>
<b>ACKNOWLEDGEMENT</b>	<b>vii</b>
<b>ABSTRACT</b>	<b>viii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xv</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xvi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Motivation of Research . . . . .	4
1.3 Research Gap . . . . .	4
1.4 Problem Statement . . . . .	5
1.5 Objectives . . . . .	5
<b>2 LITERATURE REVIEW</b>	<b>6</b>
<b>3 THEROTICAL BACKGROUND</b>	<b>9</b>
3.1 COVID 19 . . . . .	9
3.2 Neural Network . . . . .	11
3.3 Convolution Neural Network . . . . .	12
3.4 Quantum Computing . . . . .	13
3.5 Quantum Computing Basics . . . . .	15

3.6	Quantum Deep Learning . . . . .	19
<b>4</b>	<b>METHODOLOGY</b>	<b>22</b>
4.1	Data Collection . . . . .	22
4.2	System Methodology . . . . .	23
4.3	Data Preprocessing . . . . .	25
4.4	Quantum Convolution . . . . .	25
4.5	Classical Convolution Layer . . . . .	30
4.6	ReLU activation Layer . . . . .	31
4.7	Classical Fully Connected Layer . . . . .	31
4.8	Output Layer . . . . .	32
4.9	Cost Function . . . . .	33
4.10	Optimizers . . . . .	33
4.11	Model Testing in Real Quantum Devices . . . . .	34
4.12	Performance Metrics . . . . .	35
<b>5</b>	<b>RESULTS ,DISCUSSION AND ANALYSIS</b>	<b>37</b>
5.1	Experimental setup . . . . .	37
5.2	Experiments . . . . .	38
5.2.1	Experiment No.1 . . . . .	39
5.2.2	Experiment No.2 . . . . .	44
5.2.3	Experiment No.3 . . . . .	49
5.2.4	Experiment no.4 . . . . .	52
5.2.5	Experiment No 5 . . . . .	56
5.3	Comparison among Experiments . . . . .	59
5.4	Model Evaluation in Real Quantum Devices . . . . .	61
<b>6</b>	<b>CONCLUSION AND FUTURE ENHANCEMENTS</b>	<b>64</b>

6.1	Conclusion . . . . .	64
6.2	Limitations and Future enhancement . . . . .	65
	<b>REFERENCES</b>	<b>68</b>
	<b>APPENDIX A</b>	<b>69</b>

## List of Figures

1.1	Quantum Convolution Layer . . . . .	3
3.1	Quantum Computing Gates . . . . .	14
3.2	Bloch Sphere Representation of Qubits . . . . .	16
3.3	Quantum Neural Network(QNN) . . . . .	21
4.1	System Block Diagram . . . . .	22
4.3	Quantum Convolution Model . . . . .	24
4.4	Hybrid Quantum-Classical Deep Learning Model . . . . .	25
4.6	4 qubit Quantum Convolution Laye . . . . .	27
4.7	9 qubit Quantum Convolution Laye . . . . .	28
4.8	16 qubit Quantum Convolution Layer . . . . .	28
4.9	ReLU activation function graph . . . . .	31
4.10	Reservation of real quantum devices . . . . .	34
4.11	Confusion Matrix Format . . . . .	35
5.1	Training vs Validation curve of Model 1.1 . . . . .	39
5.2	Confusion matrix of model 1.1 . . . . .	40
5.3	Training vs Validation curve of Model 1.2 . . . . .	40
5.4	Confusion matrix of model 102 . . . . .	41
5.5	Training vs Validation curve of Model 1.3 . . . . .	41
5.6	Confusion matrix of model 1.3 . . . . .	42
5.7	Comparison plot for Quantum Filter vs Classical Filter in exp.1 . . . . .	43
5.8	Classification Metrics For Model 1.1,1.2 and 1.3 . . . . .	44

5.9	Training vs Validation curve of Model 2.1 . . . . .	44
5.10	Confusion matrix of Model 2.1 . . . . .	45
5.11	Training vs Validation curve of Model 2.2 . . . . .	45
5.12	Confusion matrix of Model 2.2 . . . . .	46
5.13	Training vs Validation curve of Model 2.3 . . . . .	46
5.14	Confusion matrix of Model 2.3 . . . . .	47
5.15	Comparison plot for Quantum Filter vs Classical Filter in exp.2 . . . . .	48
5.16	Classification Metrics For Model 2.1,2.2 and 2.3 . . . . .	48
5.17	Training vs Validation curve of Model 3.1 . . . . .	49
5.18	Confusion matrix of Model 3.1 . . . . .	49
5.19	Training vs Validation curve of Model 3.2 . . . . .	50
5.20	Confusion matrix of Model 3.2 . . . . .	50
5.21	Comparison plot for Quantum Filter vs Classical Filter in exp.3 . . . . .	51
5.22	Classification Metrics For Model 3.1 and 3.2 . . . . .	52
5.23	Training vs Validation curve of Model 4.1 . . . . .	52
5.24	Confusion matrix of Model 4.1 . . . . .	53
5.25	Training vs Validation curve of Model 4.2 . . . . .	53
5.26	Confusion matrix of Model 4.2 . . . . .	54
5.27	Comparison plot for Quantum Filter vs Classical Filter in exp.4 . . . . .	55
5.28	Classification Metrics For Model 4.1 and 4.2 . . . . .	55
5.29	Training vs Validation curve of Model 5.1 . . . . .	56
5.30	Confusion matrix of Model 5.1 . . . . .	56
5.31	Training vs Validation curve of Model 5.2 . . . . .	57
5.32	Confusion matrix of Model 5.2 . . . . .	57
5.33	Comparison plot for Quantum Filter vs Classical Filter in exp.5 . . . . .	58

5.34	Classification Metrics For Model 5.1 and 5.2 . . . . .	59
5.35	Training Time of all models . . . . .	60
5.36	Training Parameters of all models . . . . .	60
5.37	Accuracy of all models . . . . .	61
5.38	Avg. CNOT gate error in real quantum devices . . . . .	61
5.39	Instance of quantum circuit after transpiling . . . . .	62
5.40	Instance of quantum circuit showing frequency plot of each state . . . . .	62
5.41	Histogram Showing Model Accuracy in different real quantum devices . . . . .	63

## List of Tables

4.1	Real Quantum devices specifications . . . . .	34
5.1	Hyperparameters . . . . .	38
5.2	Experiments in this research work . . . . .	38
5.3	Trainable parameters in each layers of models in experiment 1 . . .	39
5.4	Model Summary of Model No 1.1,1.2 and 1.3 . . . . .	43
5.5	Model Summary of Model No 2.1,2.2 and 2.3 . . . . .	48
5.6	Model Summary of Model No 3.1 and 3.2 . . . . .	51
5.7	Model Summary of Model No 4.1 and 4.2 . . . . .	54
5.8	Model Summary of Model No 5.1 and 5.2 . . . . .	58
5.9	Results of all models . . . . .	60

## LIST OF ABBREVIATIONS

CNN	: Convolution Neural Network
COVID	: Coronavirus Disease
PCR	: Polymerase Chain Reaction
WHO	: World Health Organization
QCNN	: Quantum Convolution Neural Network
NISQ	: Noisy Intermediate Scale Quantum
HOG	: Histogram Oriented Gradients
AUC	: Area under Curve
GPU	: Graphics Processing Unit
CPU	: Central Processing Unit
NAAT	: Nucleic Acid Amplification Test
QPU	: Quantum Processing Unit
SVM	: Support Vector Machine
SARS	: Severe Acute Respiratory Syndrome
MERS	: Middle East Respiratory Syndrome
ARDS	: Acute Respiratory Distress Syndrome
DNN	: Deep Neural Network
ML	: Machine Learning



# CHAPTER 1

## INTRODUCTION

### 1.1 Overview

The World Health Organization (WHO) declared a pandemic when the coronavirus (COVID-19), which first appeared in China in December 2019, rapidly spread around the world [1]. During an outbreak, being able to identify COVID-19 in a patient who is affected is crucial. It is anticipated that a reverse transcription real-time fluorescence polymerase chain reaction (RT-PCR) test for nucleic acid amplification (NAAT) of the respiratory tract or blood samples would produce positive findings to identify COVID-19 [2]. However, because to the low viral load in the early stage and existing clinical experience, the detection rate and sensitivity are limited. Negative outcomes will inevitably follow as a result. If not, it can only have positive or negative outcomes. It is impossible to keep track of the infection's intensity and development. After collecting the patient's sample, it might take up to a day or longer to get the results. Chest X-rays are used to identify whether a person has COVID Pneumonia or is healthy. After collecting the patient's sample, it might take up to a day or longer to get the results. Chest X-rays are used to identify whether a person has COVID Pneumonia or is healthy.

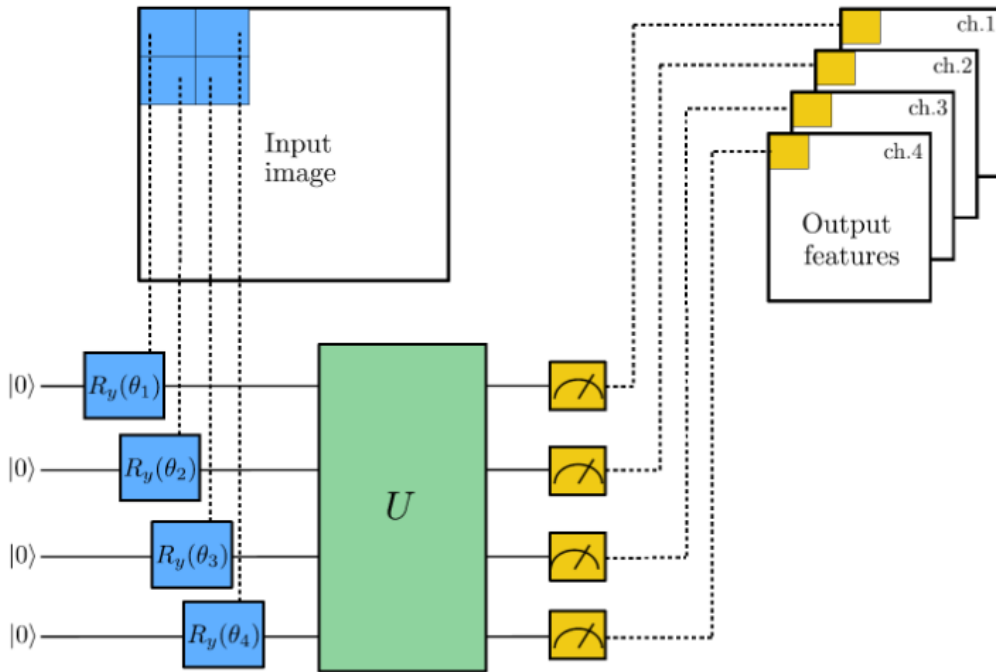
Medical image classification is important field of image classification. It assists the doctors or professionals to make decision about the disease with the help of medical images. Collecting large dataset of medical images require a huge expertise and medical image collection is related to security issue and time-consuming. Many machine learning and image classification model have been developed for medical image classification. But they are complicated and quite expensive. In the medical images the model have to extract small and irregular patterns which a different from the real world image datasets. Traditional machine learning algorithms such as SVM(Support Vector Machine)were used for biomedical image classification but they were not able to extract reasonable feature and performance lower compared to other model. So CNN is regarded as the best feature extractor for the medical

image classification. Unlike real world images the medical images are limited ,inorder to develop a efficient model we need a lot of training dataset for building the model from the scratch.We also a use pre-trained model transfer learning approach to build a good model by transfer maximum weights of the pre-trained model. Ensemble method are also used for extraction of feature by different models and combine them for classification. But feature extraction by classical deep learning model is a complex process.It requires large computational complexity and often time consuming.So reseachers have been focusing in development of image classification model that performs well on the small datasets and computationally efficient.

Quantum Computing is trending computing technique now a days.Quantum computing has been superior over classical over certain tasks.However ,it could not replace the classical computer in all aspects. Combination of classical computing and quantum computing may be useful in many tasks. In recent years, there have been major developments in the quantum computing field. Quantum Computing is a hold at the crossroads of computer science and quantum mechanics. It is more powerful than the old computer as it uses quantum bits with two identical regions of old pieces (either 0 or 1). These quantum pieces can have both values at the same time and are commonly referred to as the top provinces. A computer that I can use these quantum pieces are a quantum computer. Quantum pieces are also called qubits. Provincial assets bring significant benefits to the industry. In-depth reading is a form of machine learning that works with neural networks to mimic the functioning of the human brain. It can be used for a variety of tasks image classification, image classification, and more. With advances in the field of deep learning, convolutional neural networks (CNN) have been very effective. Traditional machine learning models. An in-depth reading of the model works best when given a large database.

With the quantum supremacy of Shor's algorithm [3] to find prime factors of large number which would takes years for the classical computer to compute.Grover's algorithm[4] is another algorithm that uses the quantum computing mechanism to search a instance from large number of datasets.The researches now are been focused what quantum computer offer to deep learning methods.Several quantum machine learning algorithms such Quantum Support Vector Machine [5],Quantum

Variational Eigen solver[6] had performed well in machine learning tasks. In this noisy NISQ era there is availability less number of noise less qubits, so we have to develop algorithm relative to this era which could be useful for the nearby quantum devices. From the perspective of transfer learning several experiment have been done by extracting the features from the classical computer and classification done by the quantum computer[7][8][9]. Cong. et.al[10] gave the concept of quantum neural network which used to classify the datasets based on parameterized quantum circuit. Henderson .et al [11] also gave the concept of quantum convolutional i.e. quantum convolutional neural network that will work similar to classical convolutional layer for extracting the feature from the given images. As the dataset increases ex-



**Figure 1.1:** Quantum Convolution Layer

ponentially it is difficult to hand by a classical computer. The quantum convolution filter is convoluted over the subsection of an image to form the convoluted image. Instead of convoluting the entire image, we divide the image into subsections and convolute the entire image. Due to the availability of a limited number of qubits, we down sample the original image and apply the convolution filter. After the convolution operation, we get a new feature map and apply the quantum pooling operation to reduce the complex structure of the quantum circuit. We apply different layers of convolution and pooling layers to get a reduced feature map and pass

it to the fully connected classical layer to classify the image. Figure 1.1 shows that could also use the quantum computer as a feature extractor and use the classical computer to classify the images. A Quantum computer could extract more details than a classical computer. Further, we could also use a full quantum computer for feature extraction and classification purpose. However, data sets containing medical, satellite, and real-time images are limited to special permissions. As the number of data samples is limited to those in databases, in-depth learning algorithms may not produce the expected results. This encourages the use of transfer learning which is a clever way of passing information obtained from one place to another. It is a recent development in the context of the machine learning domain but is used almost always in real-time situations.

## **1.2 Motivation of Research**

The medical data are very sensitive, for the proper diagnosis, the medical image classification must provide accurate and faster results. Classical computers have complexity in training and don't perform well in big data. The phenomenon of quantum parallelism along with the interference and entanglement feature of quantum computers could speed up the processing as they perform calculations at multiple states. However, fully high processing quantum computers are not available due to limited numbers of qubits in NISQ era. We could use combination of both quantum computers and classical computers for better performance. Quantum computing is trending topic nowadays, and researchers are focused on developing better quantum algorithms.

## **1.3 Research Gap**

The rapid rise of COVID-19 has made a serious problem for people all over the world. So, it is very necessary to quickly and accurately identify the person infected and isolate them to prevent from future spreading of disease. Identification of COVID-19 by application of image processing uses mainly Chest X-ray and CT-Scan images. The deep learning algorithms can identify the pattern in COVID

cases and normal cases and classify them. Many types of research have been done in Classical Neural Network and deep learning for this disease prediction. Ensemble and fusion methods are also done for better performance. Few quantum algorithms have also been used to enhance the classification task. Several hybrid classical-quantum models have improved the performance. A lightweight classification full quantum neural network have also been developed for COVID detection by using CT-Scan images. However, use of full quantum deep learning model for COVID detection by using Chest-X-ray is missing. Most of the experiments are performed on small datasets. The research are mostly done on simulators rather than real quantum devices.

#### **1.4 Problem Statement**

Classical neural networks have complex structures and have more number of parameters to be calculated which increases the time and cost complexity. They contain large number of hidden layer for better feature extraction, whereas, same feature could be extracted from the quantum circuit with low numbers of parameters. So, hybrid quantum-classical model is implemented in order to improve the performance and reduce complexity. Image classification using quantum convolution is done only for small size input images for only binary class classification. Most of the researches have been done for use of only one quantum convolution layer in the hybrid model. The filters mostly used for these researches are non –trainable quantum filters.

#### **1.5 Objectives**

1. To build efficient hybrid Quantum Classical Deep Learning models for prediction of COVID-19 using Chest-X-ray images.
2. To evaluate the performance of various models on different Simulators (CPU, GPU) and real Quantum devices (QPU).

## CHAPTER 2

### LITERATURE REVIEW

The development of computer hardware has always supported the new computer innovations community. The introduction of affordable GPUs in 2010 has led to various studies of image recognition and object acquisition using in-depth learning models. Before GPUs, various handmade features such as dynamic scale features (SIFT) and histogram-oriented gradients (HOG) are designed and integrated to form a featured fund (BOF) machine learning models are common and are used for practical application of image classification. These features were based on algorithms and were biased toward the database and engineer. In-depth reading methods have helped to produce and automate image element extraction. Nowadays CNN is increasingly used for image classification, various works have been done for COVID-19 prediction by various researchers. The literature section shows the various research on COVID-19 classification using deep learning networks. Detection of this disease was based on CT images and CXR images.

In 2021 Kaur, P .et.al [12] has proposed an image classification model “COVID-Net”, to identify COVID-19 form Chest X-Ray (XR) images. This work used features extraction by pre trained model InceptionV4 architecture and use multi-class SVM classifier to distinguish among four different classes. This help the radiologist to improve their accuracy for prediction of disease. The model performed well in precision, recall and accuracy among all the previously published works. This model “COVID-Net” can be used where test kits are in short supply. Increasing the size of dataset could yield better performance of the model in future. This model would consume less time for detection of COVID-19.

Hussaina E. et.al [13] proposed a new CNN learning algorithm CoroDet for COVID-19 identification using both CT-scan images and chest X-ray. It was developed for 2 class classification (COVID and Normal), 3 class classification COVID, Normal, and non-COVID pneumonia), and 4 class classification (COVID, Normal, non-COVID viral pneumonia, and non-COVID bacterial pneumonia).It was regarded as superior

over other past model. It used 22 layer model for empirical justification. This model was used by medical technician. The future works that could be enhanced to this work was application to more medical hardware. More training datasets could increase the accuracy.

Sarki R. et.al [14] evaluated the various old neural network with the dataset and developed CNN model from scratch. 100% accuracy was achieved with binary classification and 87.50% accuracy for three classes' classification in transfer learning. Building CNN from scratch they got accuracy of 93.75%. Increasing number of classes accuracy drops in transfer learning. The dataset were collected from various sources was robust to real world scenarios. This model was useful to the rural area to classify chest related diseases where radiologist was short. This model will extend further for detection of SARS, MERS, ARDS, bacterial Pneumonia using x-ray images, and the other part consist training neural network model this dataset produced from the generative adversarial network. Increasing dataset would increase the robustness of the model.

Houssein E.H. [15] in 2022 proposed a hybrid quantum classical neural network model using variational quantum circuit to classify chest x-ray images. The dataset had 5455 chest X-ray images, 1350 COVID-19, 1350 normal, 1345 viral pneumonia, and 1400 bacterial pneumonia images. After the evaluation of this model, it showed good classification report. This model is not supposed to work well with large dataset and improving this model for big data would be future enhancement. By taking this model to real world radiologist feedback from them would improve the model performance.

Sengupta K. et.al [16] developed a faster clinical algorithm to detect COVID-19 patients using CT-Scan images. They developed full quantum machine learning model for this task. This simulation work was done in classical computer. This model improved the performance and rate of prediction. This model performed well than past classical CNN models. The model took 52 min in quantum hardware and 1h 30 min in simulator for training. This shows that QNN performed well DNN, CNN, 2D CNN with gaining accuracy by more than 2.92% with an average recall of around 97.7%. The QNN model converged faster than the other traditional models. More work could be done by integrating this model with medical devices.

Junaid M. et.al[9] proposed a framework for chest x-ray image classification based on classical to quantum transfer learning. Features were extracted from classical computer and transferred to quantum layers for classification. Extracting the best feature was challenging task. Different pretrained models were fused in serial and top most features were selected using PCA method. Results showed that this transfer model performed well than traditional classical to classical transfer model. Limitation of this work was this model was only used for classification not for segmenting the infected lung parts.

Johri S. et.al [17] performed a novel machine learning automatic model for COVID-19 detection. The feature extraction was done by two methods, Haralick and Hu moments. To classify the disease ML model was used. This model had accuracy of 92.4%, 88.24% and 87.13% in the training, validation and testing set respectively. They perform various ensemble learning mechanisms for ML algorithms. The model had accuracy of 95%. Similarly, various classification of chest x-ray using CNN from scratch and transfer learning was proposed [18][19][7][20][21].



## CHAPTER 3

### THEROTICAL BACKGROUND

#### 3.1 COVID 19

The Coronavirus disease (COVID-19) outbreak, that began in 2019 in Wuhan, China, has expanded globally, claiming more than 2.5 million lives by the end of March 1, 2021. SARS-CoV-2 virus was responsible for bringing in COVID-19. The majority of virus-infected individuals will have a mild to serious respiratory disease and will recover without the need for special care. However, some people will get serious illnesses and need to see a doctor. Serious sickness is more likely to strike older persons and those with underlying medical illnesses including cancer, diabetes, cardiovascular disease, or chronic respiratory diseases. COVID-19 can cause anyone to get very ill or pass away at any age. Being knowledgeable about the illness and the virus's propagation is the greatest strategy to stop or slow down transmission. By keeping a distance of at least one meter between people, using a mask that fits properly, and often washing your hands or using an alcohol-based rub, you may prevent infection in both yourself and other people. When it's your turn, get your vaccination, and abide by any local advice. When an infected person coughs, sneezes, speaks, sings, or breathes, the virus can spread from their mouth or nose in minute liquid particles. It's crucial to use proper respiratory technique, such as coughing into a flexed elbow, and to confine yourself to your house and rest until you feel better. In order to predict the epidemic and the efficacy of various population-wide strategies, including lockdown, social distancing, quarantine, testing and contact tracing, and media-related awareness, among others, to mitigate the spread of COVID-19, several modeling studies were carried out in the early stages of the outbreak.

- Polymerase chain reaction(PCR): This test looks for genetic material from a particular organism, like a virus. If you are infected with a virus at the time of the test, the test will reveal its presence. Even after you have stopped being

infected, the test may still be able to find viral pieces. The SARS-CoV-2 virus, which causes COVID-19, has genetic material (ribonucleic acid or RNA), which may be found in your upper respiratory samples when you do the polymerase chain reaction (PCR) test for COVID-19. Scientists employ polymerase chain reaction (PCR) technology to convert tiny quantities of RNA from specimens into deoxyribonucleic acid (DNA), which is then duplicated until SARS-CoV-2 is detected. Your test results should be available to you as soon as 24 hours after the sample is collected, but occasionally it may take a few days, depending on how quickly the sample is transported to the laboratory.

- **Rapid Antigen Test(RDT):** For quicker COVID-19 diagnosis, a new technique has just been developed and made available. Using a technique called lateral flow immunoassay, the antigen detection rapid diagnostic test (Ag-RDT) directly identifies viral proteins or antigen of SARS-CoV-2, the virus that causes COVID-19, in respiratory samples. Despite being less sensitive than rRT-PCR, this test is simpler to apply and has a quicker turnaround, making test results accessible in less than 30 minutes. It can also be done right at the point of treatment, negating the need for a biosafety level 2 (BSL2) laboratory facility. Early case detection is greatly aided by the use of Ag-RDT, which also aids in prompt and proper patient care and public health decision-making. Ag-RDT is particularly tempting to employ in places and high-risk circumstances with limited resources due to its excellent speed and other qualities. Therefore, in situations where NAAT is not accessible or when a lengthy turnaround prevents clinical relevance, WHO advises using Ag-RDT to identify SARS-CoV-2 infection.
- **Blood Sample Test:** After a patient has fully recovered from COVID-19, antibody testing, also known as serology testing, is often performed. Depending on the availability of exams, eligibility may change. A medical expert obtains a blood sample, often by pricking the patient's finger or taking blood from an arm vein. The sample is then examined to see if you have produced antibodies against the COVID-19-causing virus. These proteins, known as

antibodies, are created by the immune system and are essential for battling and eliminating the virus. If test results reveal that you have antibodies, this may indicate that you have had prior COVID-19 infections or that you have developed antibodies as a result of vaccination. You may also have some immunity if that is the case.

- **Medical Imaging Based Diagnosis:** Throughout the coronavirus disease-2019 (COVID-19) pandemic, radiography and the radiologist's function have changed. Chest computed tomography was first utilized for COVID-19 screening and diagnosis, but it is currently only recommended in high-risk patients, individuals with advanced illness, or in locations with limited access to polymerase chain reaction testing. Chest radiography is currently mostly used to keep track of disease development in hospitalized patients displaying indicators of deteriorating clinical state. Additionally, throughout the COVID-19 pandemic, several operational hurdles in the field of radiography have been addressed. The use of teleradiology and virtual care clinics significantly improved our capacity for social isolation, and both are expected to continue to play a significant role in the delivery of diagnostic imaging and patient care. As the pathophysiology of the virus is further recognized and the identification of predisposing risk factors for complication development is further established, there will be greater opportunities to use imaging for the diagnosis of extrapulmonary symptoms and complications of COVID-19 illness. The various body images used in imaging based detection are: Chest X-ray, Chest MRI, Chest CT-Scan e.t.c

### **3.2 Neural Network**

Machine learning includes a subclass known as a neural network. It is the network that mimics how the human brain functions and enables computers to identify patterns in input data and categorize them into corresponding output values. Here, the term "pattern" refers to the underlying relationship between the input data. In a neural network, there are three different kinds of layers. The first layer of the neural network, known as the input layer, is in charge of receiving inputs and

sending them to the hidden layers in the second layer. Hidden layers use activation function to provide weight to the input. The output values of the hidden layer were processed from the input. The third layer of the neural network, the output layer, then outputs the outcome.

### 3.3 Convolution Neural Network

Convolution neural networks are among the most popular neural networks. CNNs are generally used for scene identification, picture recognition, and image grouping using learnable weights and biases. CNN is made up of neurons with weights. CNN neurons receive inputs and apply a weighted sum to them. And these inputs were then sent to an activation function, which eventually generated an output. The Convolution layer, pooling/subsampling layers, and activation are CNN components. totally linked layers and layers. CNN has the advantage of requiring less resources. comparing preprocessing to other methods It has the ability to extract the concealed data information on its own.

- Convolution Layer: There are several separate filters in this layer. Every CNN filter individually processes each input picture to create corresponding feature maps. While CNN's higher-level layers extract higher-level characteristics, the first layer only extracts low-level features like edges and corners. The layers may have a large number of convolution kernels. The kernel has a direct relationship with the features. One output feature results from the convolution of one input with one kernel.
- Pooling Layer: In CNN, the pooling layer is utilized to lower the features' resolution. Noise and distortion have a little impact on the characteristics that were retrieved. In the input image, a pixel's value often tends to be comparable to that of its surrounding pixels, resulting in duplicate information in the output. A pooling layer repeatedly pulls feature value from a collection of cells. Two strategies exist for pooling. The provided input is separated into non-overlapping dimensional spaces of both kinds. The first is average pooling, which determines the average of the input value, and the second is

maximum pooling, which determines the maximum value of the input.

- **ReLU Activation Function:** More neurons are activated as the signal layer progresses from one to the next, allowing signals to travel more effectively for identification since they are tightly related with prior references. There are many functions for activation. Comparatively speaking, the Rectified Linear Unit (ReLU) is renowned for its quick training pace. A crucial component of a neural network that is non-linear in nature is the activation function. In relation to the input, this function controls whether and how much to fire up the neuron output. As the activation function, the Rectified Linear Unit (ReLU) is provided by:

$$y = \max(0, x) \tag{3.1}$$

- **Fully Connected:** CNN's last levels are completely interconnected. Each neuron in the subsequent layers is connected to the neurons in the preceding layers. Consideration is given to every conceivable path from input to output.
- **Softmax Activation Function** In the output layer of the previous example, we will utilize the Softmax activation function rather than the sigmoid function. The relative probabilities are computed using the Softmax activation function. This indicates that the ultimate probability value is calculated using the values of  $Z_{21}$ ,  $Z_{22}$ , and  $Z_{23}$ .

### 3.4 Quantum Computing

Quantum computing is the computing technology based on quantum mechanics. It can be used to solve the complex problem that are hard for the classical computers. Combination of both classical and quantum computing use to solve any kind of hard problem. Quantum computing basically uses two phenomenon for being superior over classical computing i.e. superposition and entanglement. Classical computing are based on storage units known as bit which be at 0s or 1s in quantum computer basic storage are qubits or quantum bits which can be at both the combination of state 0 or 1 at same time. Qubits are made by physical atoms such as spin of electron or orientation of photon. The qubits can we in combination

of many state at a time known as superposition phenomenon. Then the qubits can be interlinked each other i.e. changing the state of one qubits affects the state of other qubits, this phenomenon is called entanglement. Quantum computing could be superior over large number of cases. For example, finding prime factors of large number is too difficult for the classical computer, but this may be too easy for a quantum computer. There are various situations in which quantum computer may outperform classical computer. Similar to classical bit gates there are also few quantum gates that operate on qubits. They change the state of qubits from one state to other. They are single qubits, multi qubits gate. They are parameterized or non parameterized on the fly gates. They simply perform inner tensor product on state of qubits. There are 10 gates in quantum computing as listed below:

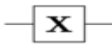


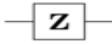

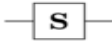
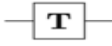
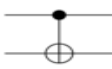
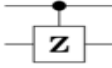



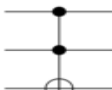
Operator	Gate(s)	Matrix
<b>Pauli-X (X)</b>	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
<b>Pauli-Y (Y)</b>		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
<b>Pauli-Z (Z)</b>		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
<b>Hadamard (H)</b>		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$
<b>Phase (S, P)</b>		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
<b>Controlled Not (CNOT, CX)</b>		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
<b>Controlled Z (CZ)</b>	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
<b>SWAP</b>	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
<b>Toffoli (CCNOT, CCX, TOFF)</b>		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Figure 3.1: Quantum Computing Gates

### 3.5 Quantum Computing Basics

1. Quantum State Representation: Data in quantum computer can be represented in the form of qubits state. The state of qubits can be represented only by 0 or 1. But, in quantum computing there is certain probability of qubits to be in state 0 and state 1. This simultaneous existence of a qubit in 0 and 1 state is known as quantum superposition. The fundamental building blocks in quantum computing is called quantum bits or qubits.

A complex vector of size 2 can be used to represent state of qubit as shown below:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (3.2)$$

where,  $\alpha$  is the probability of qubit to be in state 0 and  $\beta$  is probability of qubit to be in state 1.

State 0 in matrix representation is:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.3)$$

State 1 can be represented as:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (3.4)$$

We can also represent the qubits state by bra-ket notation. The  $\langle$ bra  $|$ ket $\rangle$  notation is also called Dirac notation. Thus any arbitrary qubit state in bra-ket notation can be represented as:

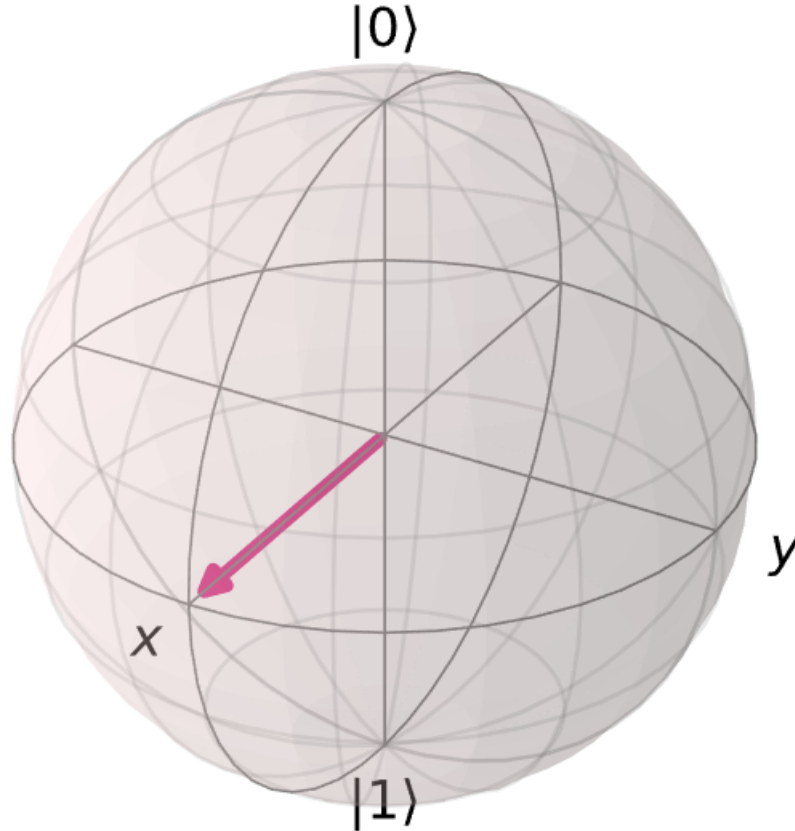
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (3.5)$$

Qubits state could be also represented in Bloch sphere. This representation consists of unit sphere, where south and north poles are placed exponentially. In

this representation we can write the general state of qubit as:

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{j\psi} \sin(\theta/2)|1\rangle \quad (3.6)$$

where,  $\theta$  and  $\psi$  lies within whole sphere without any repetitions i.e.  $\theta \in [0, \pi]$  and  $\psi \in [0, 2\pi]$ ,  $\theta$  is latitude and  $\psi$  is longitude.



**Figure 3.2:** Bloch Sphere Representation of Qubits

2. Quantum Gates: There are total of 10 quantum gates that acts on single qubits and multi-qubits. The mainly used quantum gates in this research works are:

- Hadamard Gate

Hadamard gate is a single qubit operation gate which takes the qubits in base state and turn them into superposition state. The matrix repre-



sentation of Hadamard gate is given as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3.7)$$

Hadamard operation on state  $|0\rangle$  will produce the output  $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$  and Hadamard operation on state  $|1\rangle$  will produce the output  $\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$ .

The general expression of Hadamard gate on any qubit is given as:

$$H^{\otimes n}|X\rangle = \frac{1}{\sqrt{2^n}} \sum_z (-1)^{X \cdot Z} |Z\rangle. \quad (3.8)$$

This equation can be represented by bitwise inner tensor product of X and Z.

- Rotation Gates

They are single qubits gates that rotates the state of qubits with certain parameters around the basic axes. The general expression for rotation gate is given as:

$$R(\theta, \phi) = \begin{bmatrix} \cos(\theta/2) & -ie^{-i\theta} \sin(\theta/2) \\ -ie^{i\theta} \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \quad (3.9)$$

Rotation about X-axis(RX) is obtained by putting  $\phi = 0$  in general rotation equation. It can be presented as:

$$R_x(\theta) = \exp(-i\theta X/2) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ -\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \quad (3.10)$$

Rotation about Y axis is denoted by (RY) and obtained by putting  $\phi = \pi$ , represented in matrix form as:

$$R_y(\theta) = \exp(-i\theta Y/2) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix} \quad (3.11)$$

Rotations about Z-axis is denoted by RZ and can be represented by matrix form as:

These gates are used for unitary operation of the qubits.

- Paulies X,Y and Z gates

Pauli X-gate is given in matrix form as:

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.12)$$

X-gate switches the state of  $|0\rangle$  to  $|1\rangle$  and  $|1\rangle$  to  $|0\rangle$ .

Pauli Y- gate is represented in matrix form as:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad (3.13)$$

Y-gate switches the state of  $|0\rangle$  to  $|1\rangle$  and  $i|1\rangle$  to  $-i|0\rangle$ . This gate not only changes the state of qubits but also flips the phase of the qubits.

Pauli Z gate is represented in matrix form as:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.14)$$

This gate does nothing changes to  $|0\rangle$  state but flips the sign of  $|1\rangle$  state to  $-|1\rangle$ .

3. Quantum Entanglement: Two systems away from each other behaving randomly may be correlated with each other. This phenomenon is called quantum entanglement. Entanglement is done by multi-qubit quantum gate known as Controlled Not (C-NOT). It takes two qubit states as input, one is control bit and another is target bit.

The transformation matrix of C-NOT gate is given by:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.15)$$

They are used in quantum circuit in order to introduce strong relationship among qubits.

4. Quantum Measurement: Quantum computer being probabilistic in nature cannot give accurate result as the classical computer at one measurement. So, we need to perform several execution and take average of the output value at different shots that is known as expectation value of qubits. Measurement is done in order to study the state of the qubits. The quantum states collapse to classical state during measurement. In our research work measurement is done on the parameterized quantum circuit that is used for quantum convolution. The state of quantum system are changed by the unitary operations by single and multi qubits gates. The Pauli Z gate is used for measurement of quantum state in our research work.

The general expression of expectation value of  $\phi$  state is given by the equation:

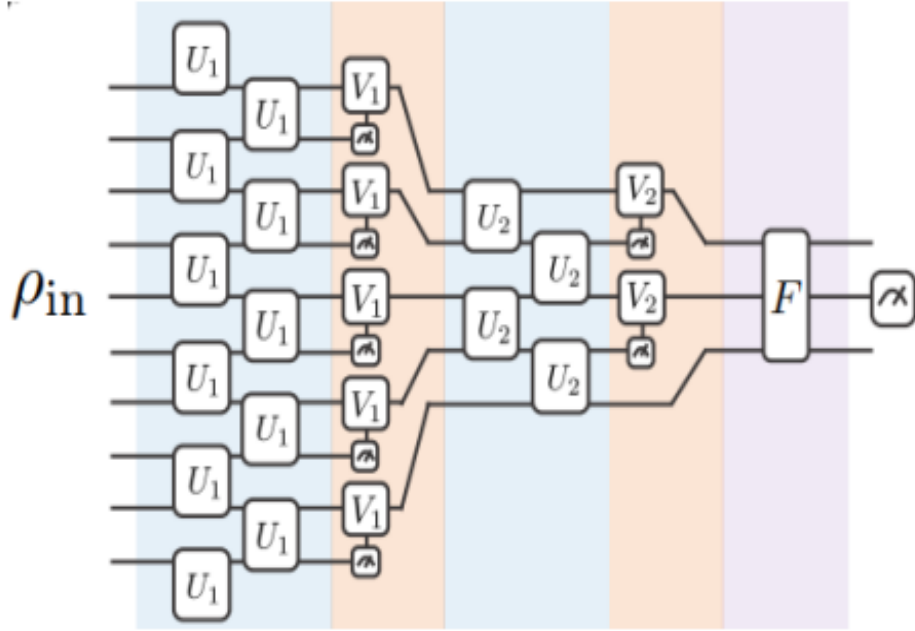
$$\langle Z \rangle = \langle \psi | Z | \psi \rangle \quad (3.16)$$

### 3.6 Quantum Deep Learning

Significant advancements in the domains of deep learning and quantum computing have been made during the previous few decades. Recently, quantum deep learning and quantum-inspired deep learning approaches have been developed as a result of the growing interest in research at the intersection of the two domains. Benefiting through the advantages of quantum computing shown by Shor's and Grover's algorithm researchers are focused on what would quantum computing offer in the field of machine learning and deep learning. Such a complex architecture in CNN could be replaced by certain quantum circuit or not. Various architecture has been

proposed for quantum deep learning.

- Quantum Neural Network: Cong iris et.al[10] proposed a new quantum computing algorithm similar to Convolutional Neural network. It only used  $O(\log(N))$  trainable parameters for  $n$  qubits systems. This was successfully implemented and training that could be useful for near by quantum devices. This model was made by the fusion of Multi-scale Entanglement renormalization ansatz (MERA) along with the quantum noise correction. They firstly recognized 1D symmetry-protected topological phases. They further used quantum error correction scheme for the particular model. Quasi-local unitary operation is performed on the input states of the qubits as the convolutional part. For the pooling parts the measurement and unitary operations is performed on the near by qubits. This helps in reducing the dimensional and provides non-linearity. The final measurement is recorded as the output of the model. The random unitary parameters similar to the learned parameters in classical neural network. This model mainly focuses on the Quantum phase recognition (QPR) and quantum error correction (QEC).



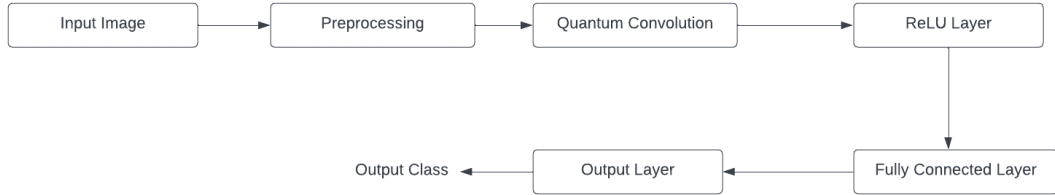
**Figure 3.3:** Quantum Neural Network(QNN)

- **Quantum Neural Network:** Henderson et.al [11] proposed a new layer from quantum computing which was similar to classical computing. They introduced this layer at several parts of the image classification model and observed the performance. The input image was convoluted by section by section similar to the classical convolution. Then the encoded quantum states are fed into the random encoder decoder and variational circuit and output was obtained in each channel of the feature map. After the observation they observed that the introduction of a quantum layer in the hybrid model improved the accuracy and training stability.
- **Quantum RNN:** The major advances going on in the field of quantum deep learning have given rise to the development of a quantum model of recurrent neural networks. Hibat-Allah et al. [22] gave a novel algorithm in the quantum version of RNNs that uses a variational wave function to learn the approximate ground state of a quantum Hamiltonian. An iterative approach of RNNs was proposed by Roth [23] for simulating the bulk quantum models through translations of lattice vectors to the RNN time index.

## CHAPTER 4

### METHODOLOGY

The block diagram of research methodology are given as:

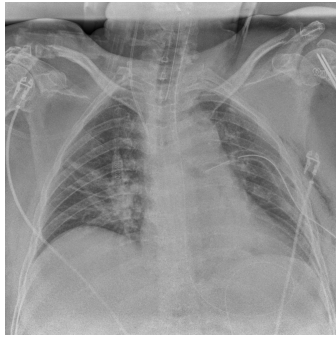


**Figure 4.1:** System Block Diagram

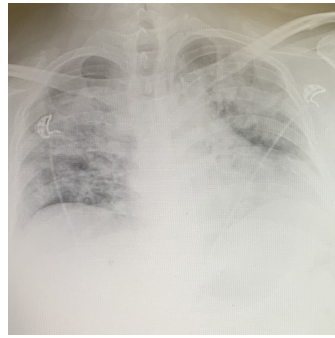
#### 4.1 Data Collection

There was no single dataset of X-ray samples related to COVID-19 so dataset was collected from multiple sources.1401 COVID-19 samples we recorded from GitHub repository, Radiopaedia , Italian Society of Radiology (SIRM) , data repository websites –Figshare.Further 912 images were collected from Mendeley instead of using explicit data expansion techniques.2313 Pneumonia cases were collected from Kaggle. Finally the dataset was organized into 3 folder (covid, pneumonia, normal) containing chest X-ray posteroanterior (PA) images.Out of 6426 images ,576 were of Covid,1577 of normal and 4273 cases were of pneumonia. The dataset was then splitted into 70% and 30% for training and validation set respectively.Again the same validation set was used for final model testing.

Classes	Training Set	Testing Set
COVID-19	403	173
Normal	1103	474



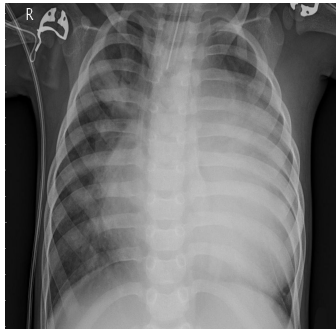
(a) COVID case I



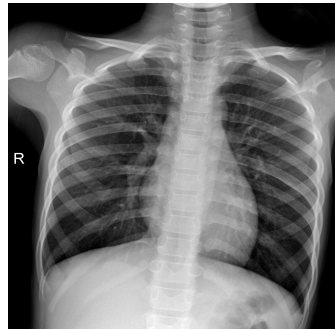
(b) COVID case II



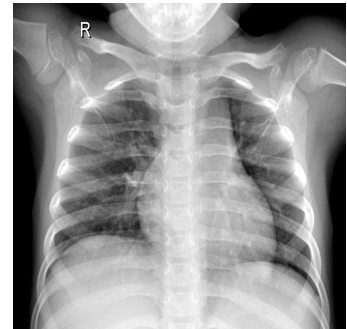
(c) Pneumonia Case I



(d) Pneumonia Case II



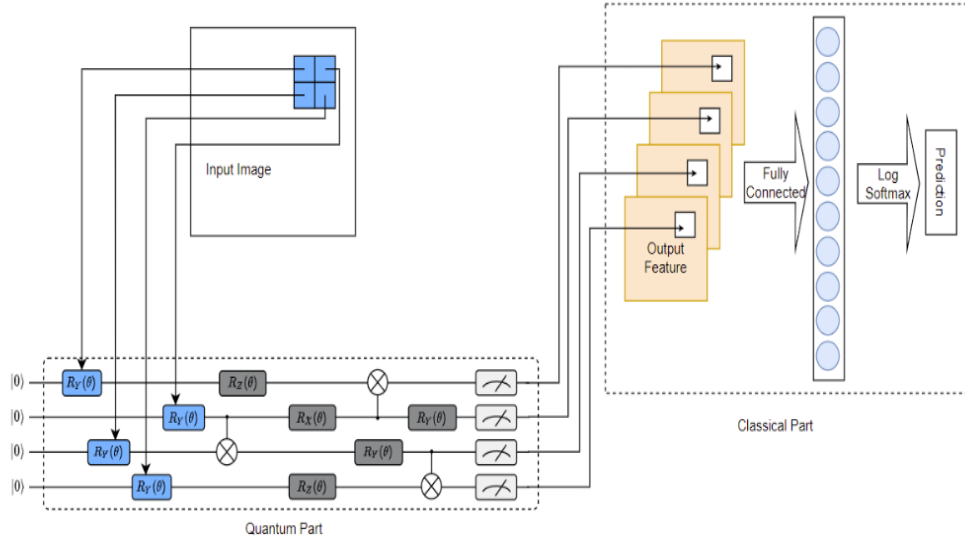
(e) Normal Case I



(f) Normal Case II

## 4.2 System Methodology

Quantum convolution for feature extraction was main focus in this thesis. So the feature were extracted from the different qubits of the quantum computer. Since we have limited number of unnoisy qubits we could not feed the whole classical image into quantum computer. So we have used stripe method for feeding the image into the circuit. Various gates were applied which changes the state of the qubits that produces non-linearity. We have also used the phenomenon of superposition and entanglement in order to get benefit from quantum computing. Since, quantum convolution layer have less number of training learnable parameters it could be fitted in between any stage of image classification pipeline in order to reduce computational complexity. The experiment performed so far had used parameterized quantum circuit for feature extraction and classification is performed by classical fully connected layer.



**Figure 4.3:** Quantum Convolution Model

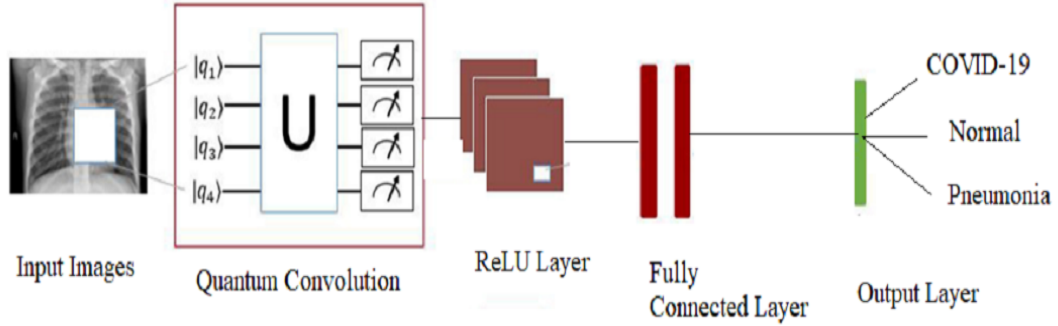
The various steps performed in this works are mentioned below:

1. The original images were resized into  $28*28, 32*32$  and  $64*64$  pixels respectively as accordance to the requirement of the experiment.
2. The training images were then normalized and argumentation is done for the preprocessing part.
3. During the quantum convolution part,  $n*n$  strips is taken from the input Images and same size quantum qubits are taken. Initially the qubits are in base state and they are taken to certain state as accordance to the parameterized operation on the qubits based on respective pixels value. Then certain unitary operation is performed on the qubits and the final value is measured. Then the measured value is mapped in respective channels of feature map. Then this process is repeated for entire image taken strip by strip.
4. Then Relu activation is applied after the quantum convolution part in order to introduce non-linearity.
5. The feature map obtained after the quantum convolution is flatten by passing it to the fully connected layer.
6. The single dimensional feature value then is passed to the output layer



which predicts the classes based on probabilities using the softmax activation function.

- The weights of fully connected layer and parameters of trainable quantum filter are fine-tuned using optimizers and loss functions. The final stable model is then used for classification.



**Figure 4.4:** Hybrid Quantum-Classical Deep Learning Model

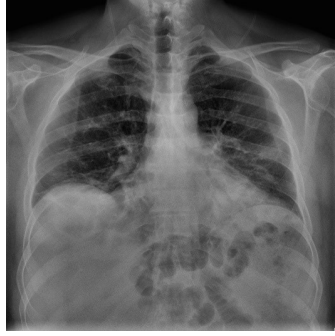
### 4.3 Data Preprocessing

The datasets were collected from different sources and of different patients. So orientation of the lungs section was not regular. So of the image were noisy and the lungs part was not clearly visible. So noisy dataset were filtered and certain image were cropped to make the lungs exact part visible. For the faster convergence of the training model the training dataset was normalized with standard deviation and mean of the training dataset .i.e.  $\text{std} = [0.4143, 0.4143, 0.4143]$ ,  $\text{mean} = [0.5160, 0.5160, 0.5160]$ .

Random resized cropping and random horizontal flip was done as a part of data augmentation as there were less data samples for deep learning.

### 4.4 Quantum Convolution

Quantum convolution is analogous to classical convolution where expectation value of each pixels are mapped to respective channel. Similar to classical convolution we slide the quantum kernel over the patches of image to obtain the feature map.



(a) Image before Normalization



(b) Image after Normalization

Quantum convolution is simply the operation of various quantum gates on the initial value of qubits which are encoded as the classical pixel value. The various steps in quantum convolution are:

1. State Preparation: Initially all the qubits are in  $|0\rangle$  state and we apply Hadamard gates to individual qubits for taking them to superposition state. Hadamard gate is one qubit that will put qubits in superposition state. If the qubit is in  $|0\rangle$  then the state will become:

$$|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (4.1)$$

And if the qubit is in  $|1\rangle$  state then the state will become:

$$|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (4.2)$$

In quantum computing logic gates can be described using matrices. The matrix associated with the Hadamard gate is:

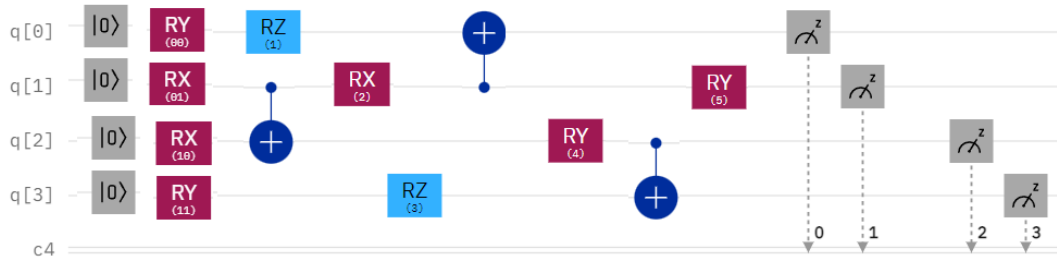
$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4.3)$$

We take pixels value from top left corner of size  $n \times n$  and feed their values in individual qubits for state preparation. For classical data conversion to quantum data we use here rotation encoding. This encoding mechanism rotates the qubits about angle value of pixels around y-axis. We use rotation

gate RY gate for this purpose. Rotation about Y-axis helps in both bit flip and phase flip. The operation of the RY gate can be described using the following matrix:

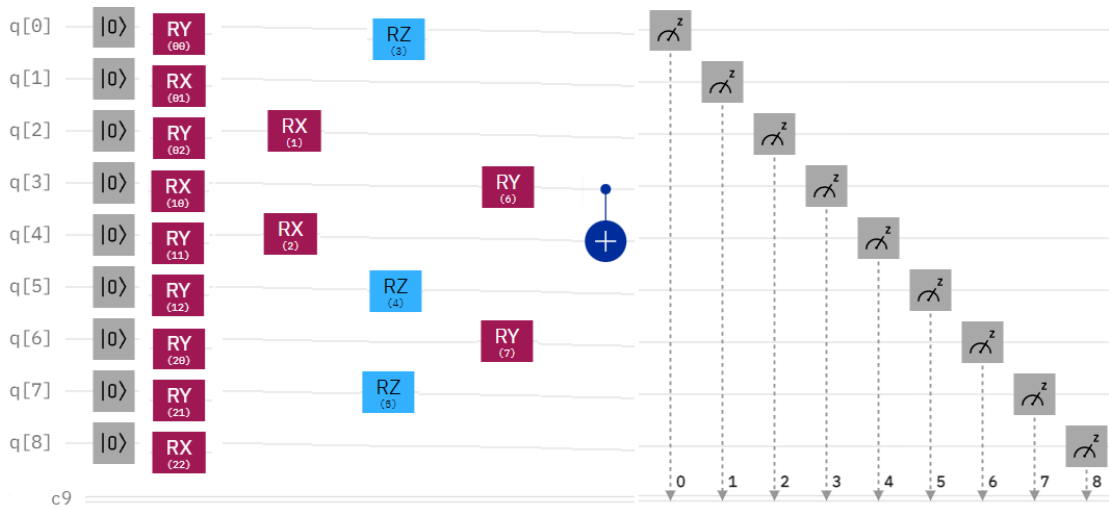
$$RY = \begin{bmatrix} \cos/2 & -\sin/2 \\ \sin/2 & \cos/2 \end{bmatrix} \quad (4.4)$$

2. Parameterized Quantum Circuit:Parameterized quantum circuit is the feature extraction part of this research .It consists several one qubits gates and multi qubits gates for alternating the state of the qubits.Here we have taken three filters for our experiment.For 2\*2 filter we have taken 4 qubits ,3\*3 filter we haven taken 9 qubits and 4\*4 filter we have taken 16 qubits.After the state preparation certain unitary operation are applied.The three quantum circuit used for feature extraction are as follows: The four channel quan-



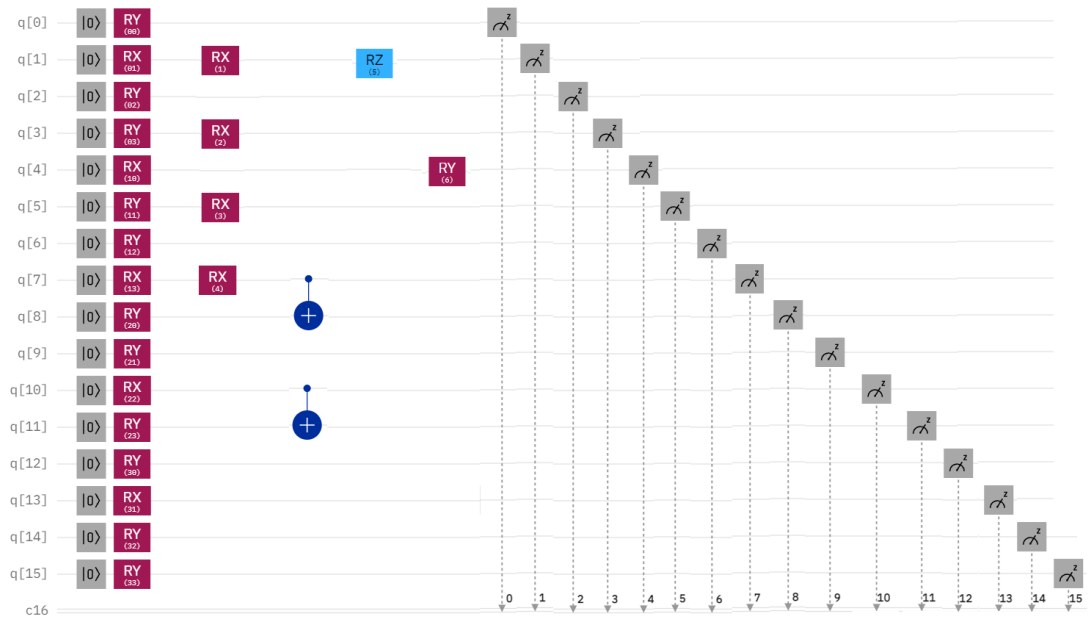
**Figure 4.6:** 4 qubit Quantum Convolution Layer

tum convolution filter consists of 8 random parameters in the four qubit system. Firstly RZ gate is applied on qubit 0 on certain random parameters  $\theta_1$ , then entanglement is performed on qubit 1 and qubit 2 in which control bit is qubit 1 and target bit is qubit 2. Then, RX gate is applied in qubit 1 about parameter  $\theta_2$ . RZ gate is applied on qubit 3 about random parameters  $\theta_3$ . CNOT gate is applied between qubit 0 and qubit 1 where qubit 1 is control bit and qubit 0 is target bit. Then RY gate is applied across qubit 2 about angle  $\theta_5$ . Then CNOT gate applied in qubit 2 and qubit 3 where qubit 2 is control bit and qubit 3 is target bit. Finally, RY gate is applied on random parameter  $\theta_6$ . For 9 channel quantum circuit we taken 9 qubits for 3\*3 strips for input image. After the state preparation we apply RX to qubit 2



**Figure 4.7:** 9 qubit Quantum Convolution Layer

for random parameter  $\theta_1$ . Then RX gate is again applied to qubit 4 at angle  $\theta_2$ . RZ rotation gate is applied to qubit 0, qubit 5 and qubit 7 at random angle  $\theta_3, \theta_4, \theta_5$  respectively. RY gate is applied to qubit 3 at angle  $\theta_6$  and qubit 6 at an angle  $\theta_7$ . Finally CNOT gate is applied to qubit 3 and qubit 4 where qubit 3 is control bit and qubit 4 is target bit. The 16 channel quantum



**Figure 4.8:** 16 qubit Quantum Convolution Layer

circuit also contains 8 random parameters for feature extraction. There are six parameterized gate and 2 on the fly entanglement CNOT gates. After the state preparation RX gate is applied to qubit 1 at random parameter  $\theta_1$ . RX

rotation gate is applied to qubits 3 ,5 and 7 at random parameters  $\theta_2, \theta_3, \theta_4$  respectively. Then multiple times entanglement is performed between qubits 1 and qubits 8 ,qubits 10 and qubits 11. Then rotation is performed about Z axis using RZ gate about random angle  $\theta_5$  for qubit 1. Finally rotation about Y-axis is done for qubit 4 about random angle  $\theta_6$ .

The change in state in one qubit changes the state of other which makes quantum computer more efficient to learn hidden patterns among images. The equation for the rotation of qubits about axis are:

$$R_x(\alpha) = \exp(-i\alpha X/2) = \begin{pmatrix} \cos(\frac{\alpha}{2}) & -\sin(\frac{\alpha}{2}) \\ -\sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) \end{pmatrix} \quad (4.5)$$

$$R_y(\alpha) = \exp(-i\alpha Y/2) = \begin{pmatrix} \cos(\frac{\alpha}{2}) & -\sin(\frac{\alpha}{2}) \\ -\sin(\frac{\alpha}{2}) & \cos(\frac{\alpha}{2}) \end{pmatrix} \quad (4.6)$$

$$R_z(\alpha) = \exp(-i\alpha Z/2) = \begin{pmatrix} e^{-0.5i\alpha} & 0 \\ 0 & e^{0.5i\alpha} \end{pmatrix} \quad (4.7)$$

The CNOT gate is a multi-qubit gate that consists of two qubits. The first qubit is known as the control qubit and the second is known as the target qubit. If the control qubit is  $|1\rangle$  then it will flip the target qubit state from  $|0\rangle$  to  $|1\rangle$  or vice versa. CNOT gate is used for entanglement purpose. The CNOT gate operation is described by the following matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.8)$$

3. Measurement: Pauli Z gate is single qubit gate and used for measurement

purpose. It is given by matrix:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.9)$$

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1| \quad (4.10)$$

The expectation value of  $Z$  in state  $|\Psi\rangle$  is given by general formula:

$$\langle Z \rangle = \langle \psi | Z | \psi \rangle \quad (4.11)$$

Thus, the expectation value on different qubits are as follows:

$$\langle E1 \rangle = \langle \psi | I \otimes I \otimes I \otimes Z | \psi \rangle \quad (4.12)$$

$$\langle E2 \rangle = \langle \psi | I \otimes I \otimes Z \otimes I | \psi \rangle \quad (4.13)$$

$$\langle E3 \rangle = \langle \psi | I \otimes Z \otimes I \otimes I | \psi \rangle \quad (4.14)$$

$$\langle E4 \rangle = \langle \psi | Z \otimes I \otimes I \otimes I | \psi \rangle \quad (4.15)$$

#### 4.5 Classical Convolution Layer

Classical CNN layer is the first layer of classical deep learning network. It consists of many neurons that fire according to the input data and bias. CNN consists of filters or kernels that apply over the section of input images to produce the extracted feature. Feature extraction is simply the convolution operation on the input image strips and the kernel values. Then the feature map is obtained in the respective feature map after application of convolution. Taking some strips during convolution can also reduce the dimensionality of the input image. Since a lot of features have to be extracted from the input images, a lot of weights for the neurons are to be trained to produce the best feature. The number of learnable parameters in a Classical CNN layer is given by:

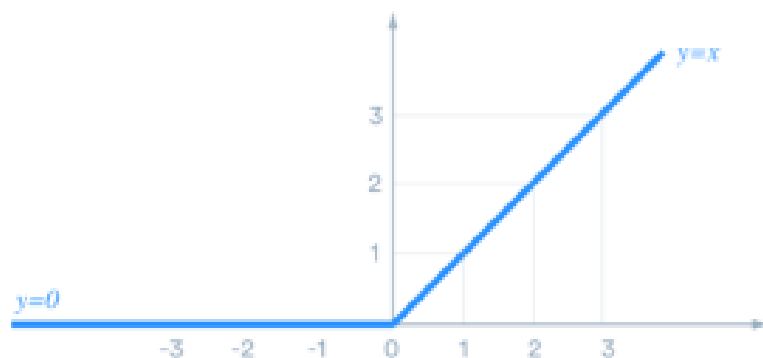
$$Parameters = (n * m * l + 1) * k \quad (4.16)$$

Where,  $n$  is kernel width,  $m$  is kernel height,  $l$  is input channel,  $k$  is output channel.

#### 4.6 ReLU activation Layer

Neuron fire based on the activation function and the input signals. There are many activation functions in neural networks. Out of them commonly used is Rectified Linear Unit (ReLU) after the convolution layer. It is because it introduces vast non-linearity that helps in detecting patterns and faster training time related to others. The ReLU activation function is mathematically given by:

$$y = \max(0, x) \quad (4.17)$$



**Figure 4.9:** ReLU activation function graph

The ReLU activation function has zero slope when the input is negative and the same slope when the input is positive. For faster convergence, faster training, and an increase in accuracy, the ReLU activation function is used in this work.

#### 4.7 Classical Fully Connected Layer

A classical fully connected layer is simply a feed forward neural network present before the output layer in a neural network. The output from the convolution layer

or pooling layer is flattened and fed into the fully connected layer. The firing of the ANN in fully connected network is as per the formula:

$$g(Wx + b) \tag{4.18}$$

Where,  $x$  — is the input vector

$W$  — is the weight from previous neuron

$b$  — is the bias of neuron

$g$  — is activation function

This layer consists of maximum number of neurons in the deep learning models and entire features are flattened and they are feed to the neurons. Therefore, computational complexity of the layer is very high as lot of weight has to be trained.

The numbers of trainable parameters in fully connected layers are as follows:

$$Parameters = (n + 1) * m \tag{4.19}$$

Where,  $n$  is the input neuron

$m$  is the output neuron

## 4.8 Output Layer

In the output layer we softmax activation function. This function converts numbers vectors in probabilities vector. Each term in the probability vector is related to the probability of occurrence of each class. The general formula for soft max activation function is:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K \tag{4.20}$$

The fully connected output layer has learnable parameter calculated based on the formula:

$$Parameters = (n + 1) * m \tag{4.21}$$

where  $n$  is the number of inputs and  $m$  is the number of outputs.



## 4.9 Cost Function

For evaluating the performance of our machine learning model cost function is very much important. It takes in the predicted out and true output and compare the results. It does not give our model from diverging from the training set by helping to fine tune the model. There are many cross function based on their working and application. This deep-learning model is multi-class image classification problem so it uses categorical cross entropy loss function.

In this multi classification model the algorithm must decide whether a object belongs to which particular class on many output classes. The categorical cross entropy calculated the loss between the predicted class and true class for an instance using the formula:

$$Loss = - \sum_{i=1}^{Outputsize} y_{ti} \cdot \log(y_{pi}) \quad (4.22)$$

Where,  $y_{ti}$  is the actual class and  $y_{pi}$  is the predicted class.

## 4.10 Optimizers

For changing and updating the random parameters such as weighs and learning rates in a CNN we use optimizers .Out of the many optimizers used in deep leaning here we use Adam( Adaptive Moment Estimation) that is based on momentum of first and second order. The principle behind this optimizer is that we don't move so fast so we go over the minima. This optimizer also keeps an exponentially decaying average of past gradients  $M(t)$ . Let  $M(t)$  be value of first moment an  $V(t)$  be value of second moment then:

$$\hat{m} = \frac{m_t}{1 - \beta_1^t} \quad (4.23)$$

$$\hat{v} = \frac{v_t}{1 - \beta_1^t} \quad (4.24)$$

Parameters update formula is:

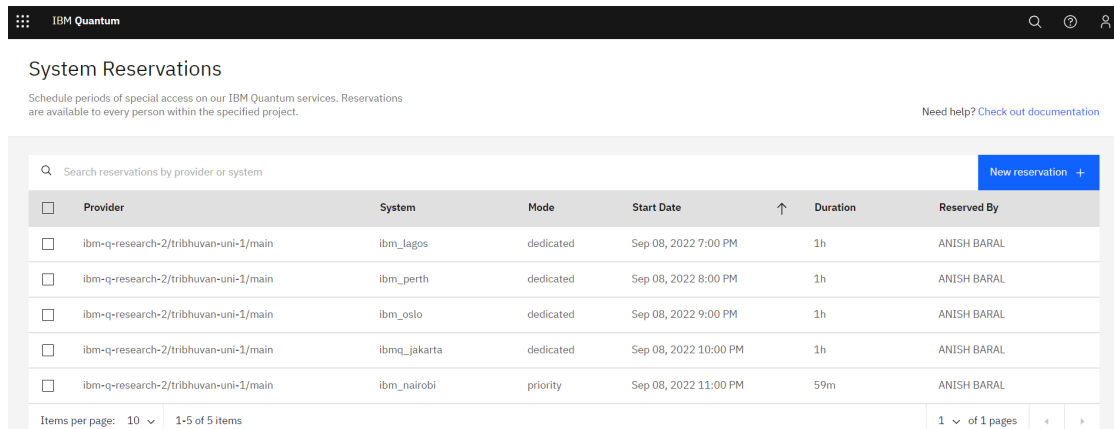
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{(\hat{v}_t) - \epsilon}} \hat{m}_t \quad (4.25)$$

This model is fast and converges rapidly with low vanishing gradient problem. But

the problem with this optimizer, is that it has large computational complexity.

#### 4.11 Model Testing in Real Quantum Devices

The Qiskit framework allows us to run the quantum circuit in real quantum devices through IBM Quantum Experience. For a normal IBM Quantum Experience user, we get a low priority account and low qubit quantum computer access. However, with a normal account, we have to stay in queue for a long time. So, it had been difficult for us to test a large number of images over the quantum computers. So, through the IBM Quantum Researchers program, we got provision for 180 minutes reservation for each five backends for 1 month. So, 1288 images were tested in these quantum devices. Jobs were sent from the Google Colab API. Since the gates we have used in our quantum circuit are not supported by all the models, the unsupported gates were transpiled before execution. Figure ?? shows the reservation of different real quantum devices in different times.



The screenshot shows the 'System Reservations' page on the IBM Quantum Experience. It features a search bar and a 'New reservation +' button. Below is a table with columns: Provider, System, Mode, Start Date, Duration, and Reserved By. The table lists five reservations for different backends: ibmq\_lagos, ibmq\_perth, ibmq\_oslo, ibmq\_jakarta, and ibmq\_nairobi, all reserved by ANISH BARAL.

Provider	System	Mode	Start Date	Duration	Reserved By
ibm-q-research-2/tribhuvan-uni-1/main	ibmq_lagos	dedicated	Sep 08, 2022 7:00 PM	1h	ANISH BARAL
ibm-q-research-2/tribhuvan-uni-1/main	ibmq_perth	dedicated	Sep 08, 2022 8:00 PM	1h	ANISH BARAL
ibm-q-research-2/tribhuvan-uni-1/main	ibmq_oslo	dedicated	Sep 08, 2022 9:00 PM	1h	ANISH BARAL
ibm-q-research-2/tribhuvan-uni-1/main	ibmq_jakarta	dedicated	Sep 08, 2022 10:00 PM	1h	ANISH BARAL
ibm-q-research-2/tribhuvan-uni-1/main	ibmq_nairobi	priority	Sep 08, 2022 11:00 PM	59m	ANISH BARAL

**Figure 4.10:** Reservation of real quantum devices

The specifications of each real quantum device are:

Real quantum Devices	Qubits	Quantum Volume	Clops	Avg. CNOT gate Error
ibmq_jakarta	7	16	2.4k	1.025e-2
ibmq_oslo	7	32	2.6k	7.025e-3
ibmq_lagos	7	32	2.7K	9.921e-3
ibmq_nairobi	7	32	2.6k	1.131e-2
ibmq_perth	7	32	2.9k	1.000e+0

**Table 4.1:** Real Quantum devices specifications

## 4.12 Performance Metrics

Classification means all about predicting the class from the input data. We build a model based on the training dataset. The random parameters of the model is fine tuned by the optimizers and loss function. At the final the performance of the model must be evaluated. There are various metrics for evaluating the performance of classification model such as confusion matrix, accuracy, log-loss and AUC-ROC curve. Among them confusion matrix is the popular one.

A confusion matrix is a table used to evaluate the performance of the model based on predicted classes and true labels. Basics terms in confusion matrix are:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Figure 4.11:** Confusion Matrix Format

True Positive(TP): Prediction is right for actual true value.

True Negative: Prediction is false for actual true value.

False Positive(FP): Prediction is true for actual false value.

False Negative(FN): Prediction is false for actual false values

Performance Evaluation from Confusion Matrix

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (4.26)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (4.27)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (4.28)$$

$$F - Measure = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (4.29)$$

## CHAPTER 5

### RESULTS ,DISCUSSION AND ANALYSIS

#### 5.1 Experimental setup

##### Hardware Requirements

Training a quantum deep learning model requires a computational power processor with GPU support. Neural network and quantum circuit gradient calculation needs a lot of parallel computing. Therefore, this experiment was done in Google Colab Pro Plus version because the runtime of the experiment was more than 24 hours and it provides background execution support which are not provided by Google Colab free version and Google Colab Pro. The specification of Google Colab Pro Plus are:

Guarantee of resources: High percentage

GPU: K80, T4 and P100

RAM: 52 GB

Runtime : 24 hours

Background Execution: Yes

Costs: 49.99\$ per month

Target Group: Heavy Users

##### Software Requirements

Coding part was done in Python 3.9. Framework used for making quantum circuit was torch quantum [24] which provides high GPU and state vector simulation. Qiskit quantum computing framework was used for running quantum circuits in real quantum devices. IBM quantum experience was used for drawing the quantum circuits. Python libraries such as numpy, pandas, pytorch, e.t.c were also used during the software development process.

## 5.2 Experiments

All experiments were performed for same numbers of epochs, learning rate, optimizer and loss function. The hyperparameters used in this work are: This research work

No.Epochs	15
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Categorical Cross Entropy

**Table 5.1:** Hyperparameters

is the extension of quantum convolution model for higher dimension images other than 28\*28. At first experiment were performed for 28 \* 28 images and experiment were further done on 32\*32 and 64\*64 size images. Three filter of size 2\*2, 3\*3 and 4\*4 were used in this experiment. Two variation of quantum filter was used i.e. trainable and non-trainable. There were total 12 experiment done in this research work. The model specification for different experiment is shown in the table below:

Exp .No	Model No.	Image Size	Filter Size	Filter Type
1	1.1	28 *28	2 *2	Classical
2	1.2	28 *28	2 *2	Non-Trainable Quantum
3	1.3	28 *28	2 *2	Trainable Quantum
4	2.1	28 *28	3 * 3	Classical
5	2.2	28 *28	3 * 3	Non-Trainable Quantum
6	2.3	28 *28	3 * 3	Trainable Quantum
7	3.1	28 *28	4 * 4	Classical
8	3.2	28 *28	4 * 4	Trainable Quantum
9	4.1	32 * 32	2 * 2	Classical
10	4.2	32 * 32	2 * 2	Trainable Quantum
11	5.1	64 * 64	2 * 2	Classical
12	5.2	64 *64	2 * 2	Trainable Quantum

**Table 5.2:** Experiments in this research work

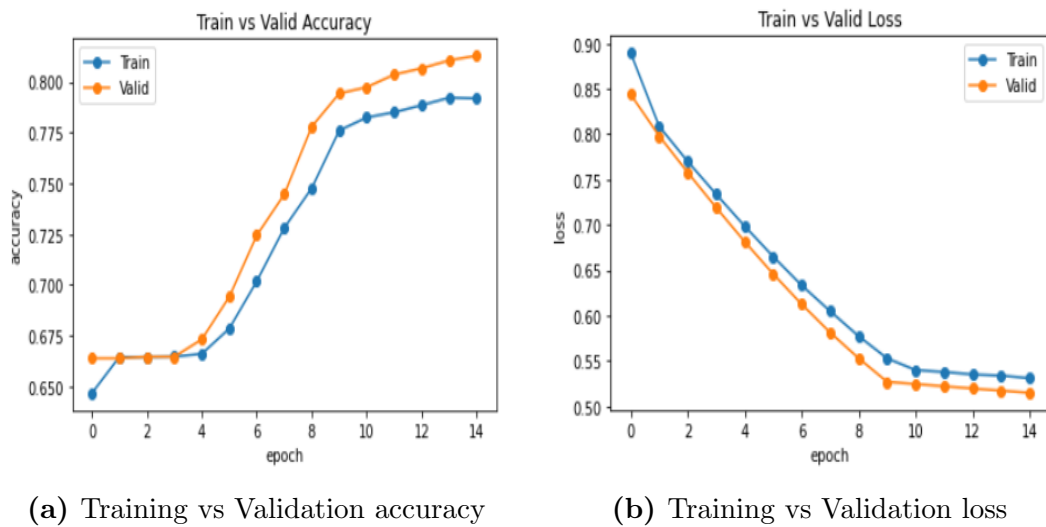
### 5.2.1 Experiment No.1

The first experiment was based on fully classical convolution model and hybrid quantum classical model for input image  $28 \times 28$  and filter size  $2 \times 2$ . First the input image were resized into  $28 \times 28$  and feed to the classical convolutional layer in model 1.1, non trainable quantum convolution layer in model 1.2 and trainable quantum convolution layer in model 1.3. After the convolution over strips of input image by  $2 \times 2$  size by convolution filter and strips of  $2, 14 \times 14$  size feature map was obtained in four channel. Then total  $14 \times 14 \times 4$  features were flattened and passed to a fully connected layer of 784 neurons. Finally, softmax activation function is applied in order to give the output in terms of probabilities of classes in the output layer. The experiment was run upto specified epochs and with specified hyperparameters.

The trainable parameters in each layer of convolution model are:

S.N.		Activation Shape	Activation Size	Model 1.1	Model 1.2	Model 1.3
1	Input Layer	$(28 \times 28 \times 1)$	784	0	0	0
2	Classical Convolution( $f=2, s=2$ )	$(14 \times 14 \times 4)$	784	20	0	5
3	Fully Connected Layer	$(784, 3)$	784	2352	2352	2352
4	Softmax	$(3, 1)$	3	3	3	3
Total:				2375	2355	2360

**Table 5.3:** Trainable parameters in each layers of models in experiment 1



**Figure 5.1:** Training vs Validation curve of Model 1.1

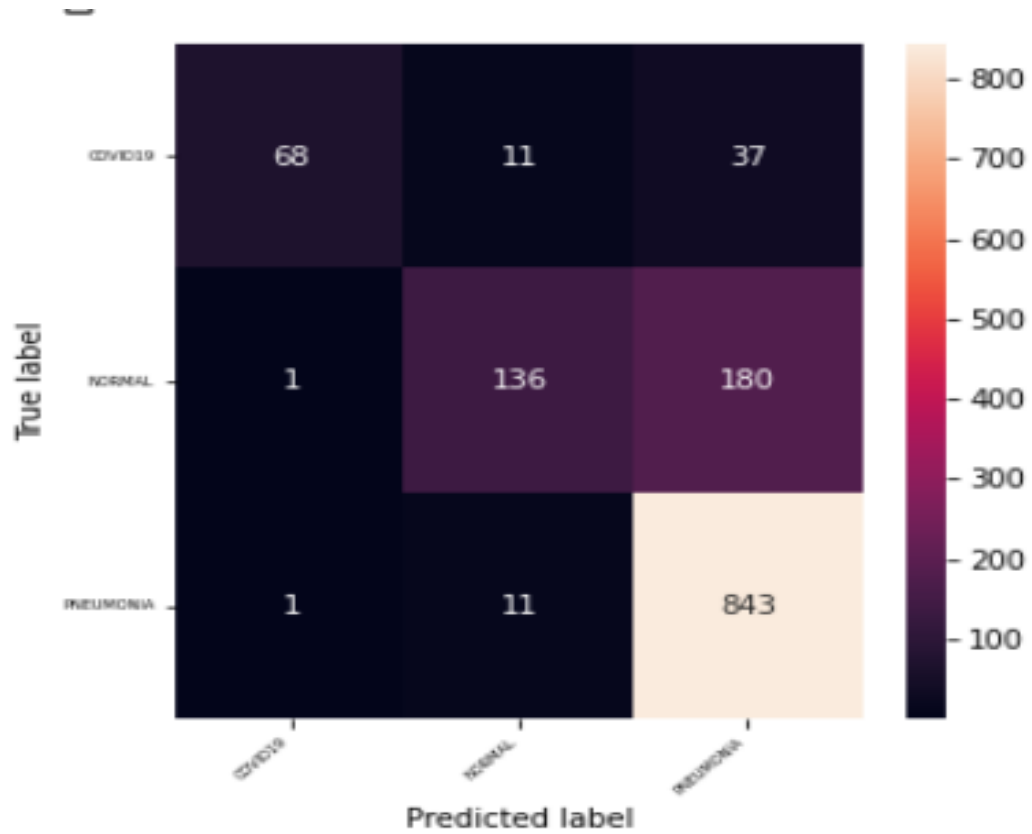
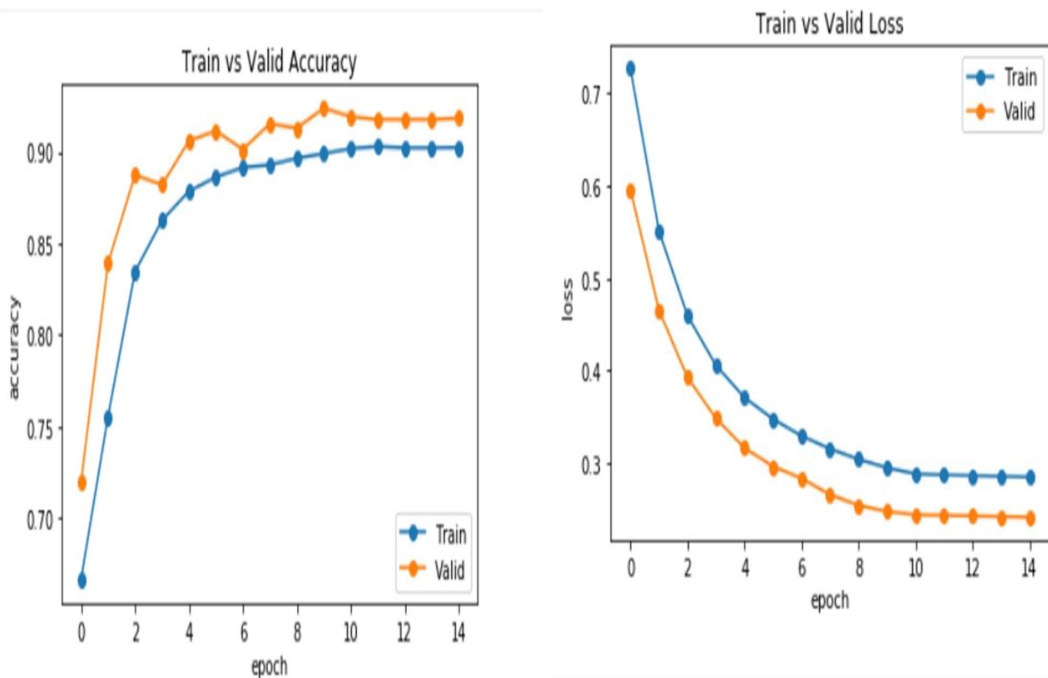


Figure 5.2: Confusion matrix of model 1.1



(a) Training vs Validation accuracy

(b) Training vs Validation loss

Figure 5.3: Training vs Validation curve of Model 1.2



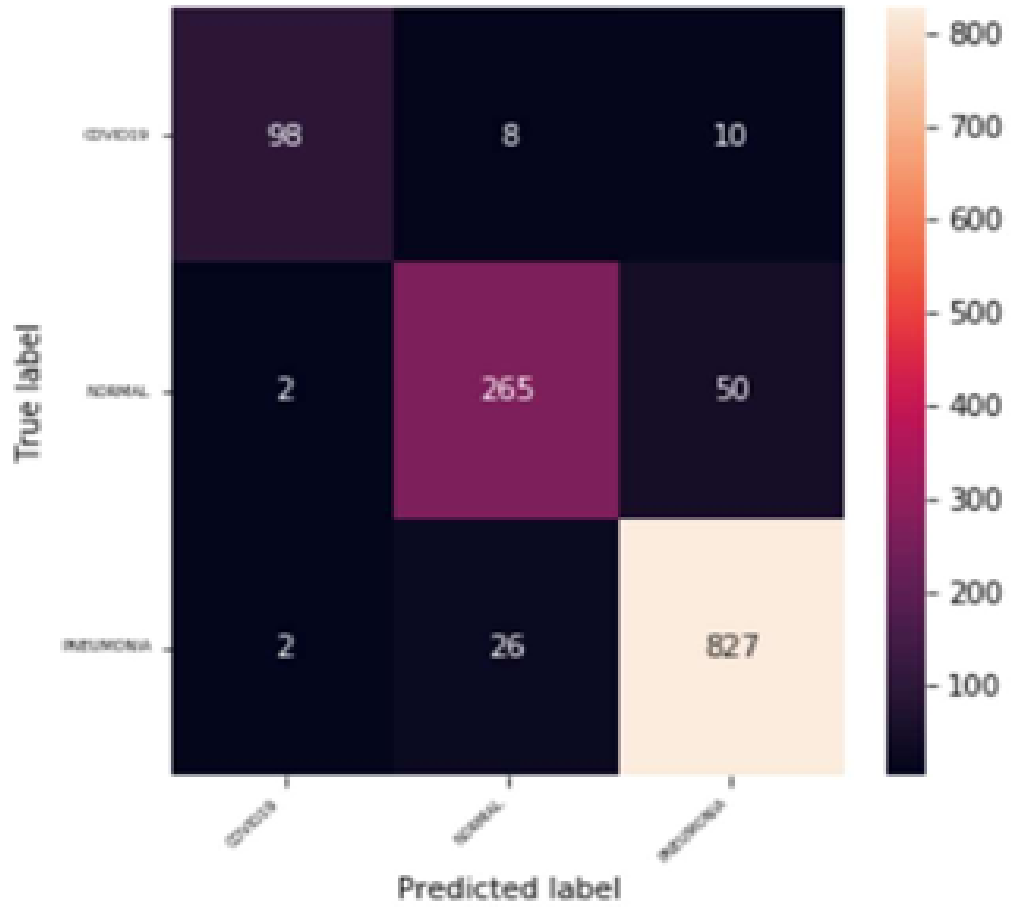


Figure 5.4: Confusion matrix of model 102

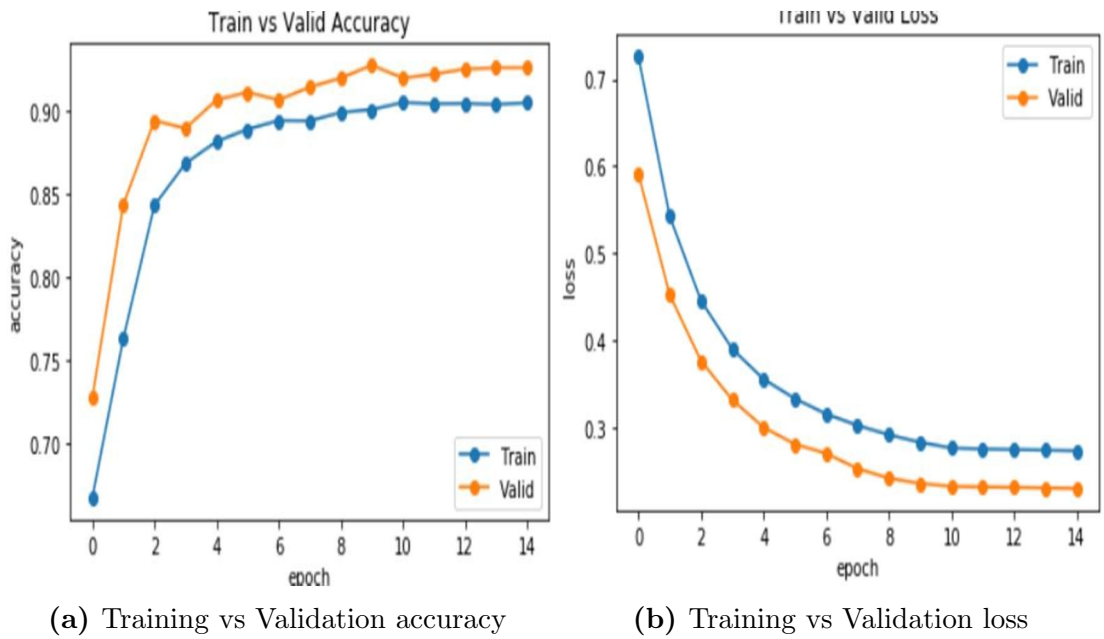
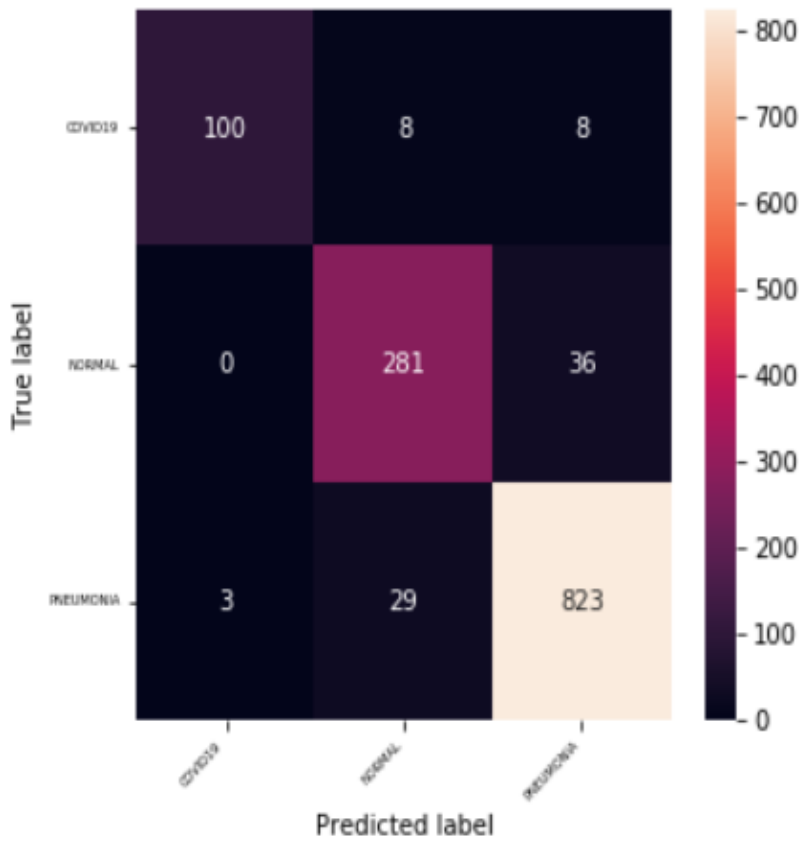


Figure 5.5: Training vs Validation curve of Model 1.3



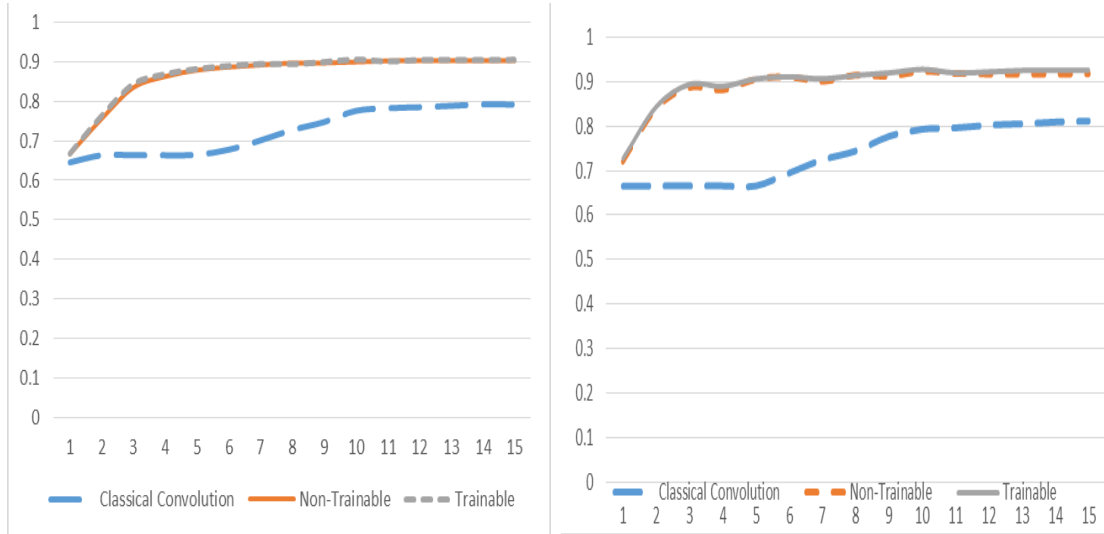
**Figure 5.6:** Confusion matrix of model 1.3

From the training accuracy and validation accuracy curve for the three models in this experiment we could find that both training accuracy and validation accuracy of the quantum convolution layer was higher. The accuracy curve of quantum trainable and quantum non-trainable was quite similar. The accuracy curve has converged from epoch 2 in the quantum-classical model the model has converged only from 10th epoch. The best accuracy for model 1.1, model 1.2 and model 1.3 were 0.8129, 0.9289 and 0.9262 respectively. Similarly in the loss curve also the loss of quantum convolution is better than the classical convolution model. The loss curve of trainable and non-trainable filter were similar. The loss curve for quantum filter converge from epoch 4 and epoch 10 for classical filter. The best train loss for model 1.1, model 1.2 and model 1.3 were 0.5311, 0.2851, 0.2737 respectively. The classical model in this experiment had low accuracy and high loss that it makes big errors in most of the data. Accuracy is increased and loss is decreased in case of quantum filter. The quantum filter outperformed the classical filter in terms of

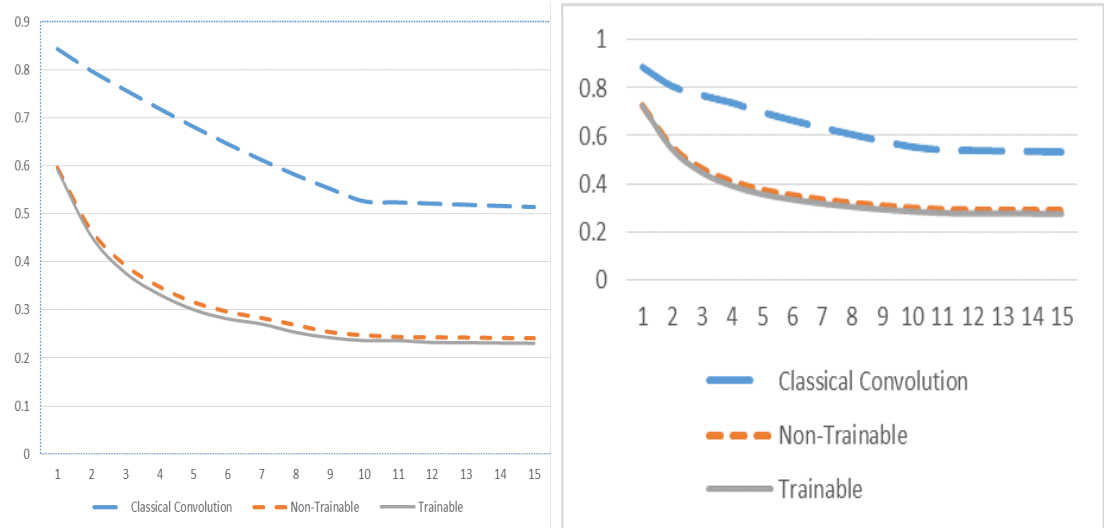
test accuracy, precision, recall and F1-score. The summary of classification of each model are listed below:

Model No.	Training Time(Min)	No. of. Trainable Parameters	Accuracy	Precision	Recall	F1-Score
1.1	36m 22s	2375	0.7931	0.5833	0.79	0.6233
1.2	142m 39s	2355	0.9239	0.9266	0.9266	0.9033
1.3	212 m 48s	2358	0.9281	0.91	0.93	0.92

**Table 5.4:** Model Summary of Model No 1.1,1.2 and 1.3

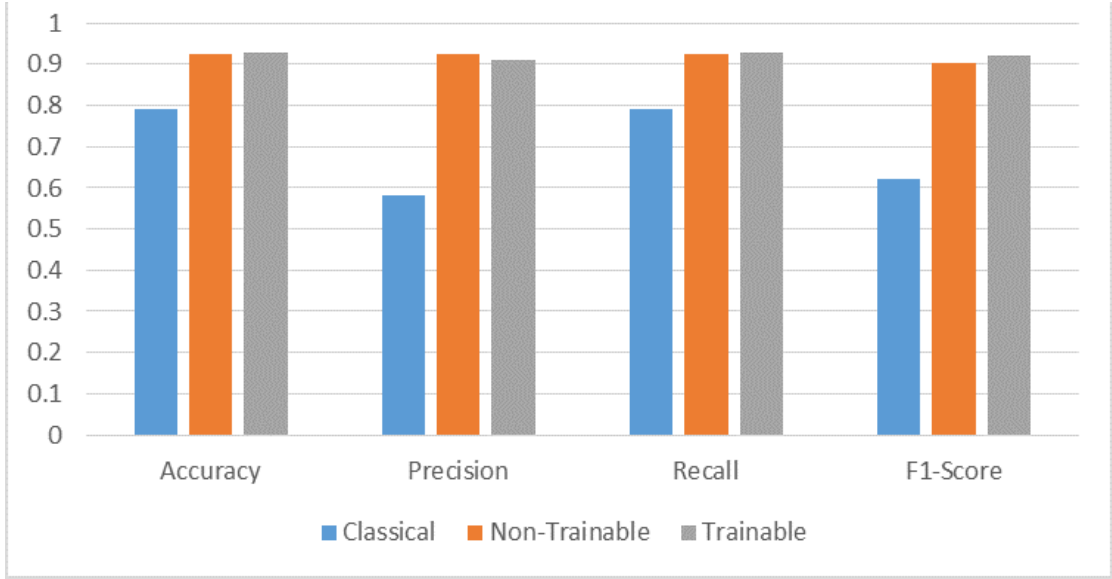


(a) Training accuracy of Quantum Filter vs Classical Filter (b) Validation accuracy of Quantum Filter vs Classical Filter



(c) Training Loss of Quantum Filter vs Classical Filter (d) Validation Loss of Quantum Filter vs Classical Filter

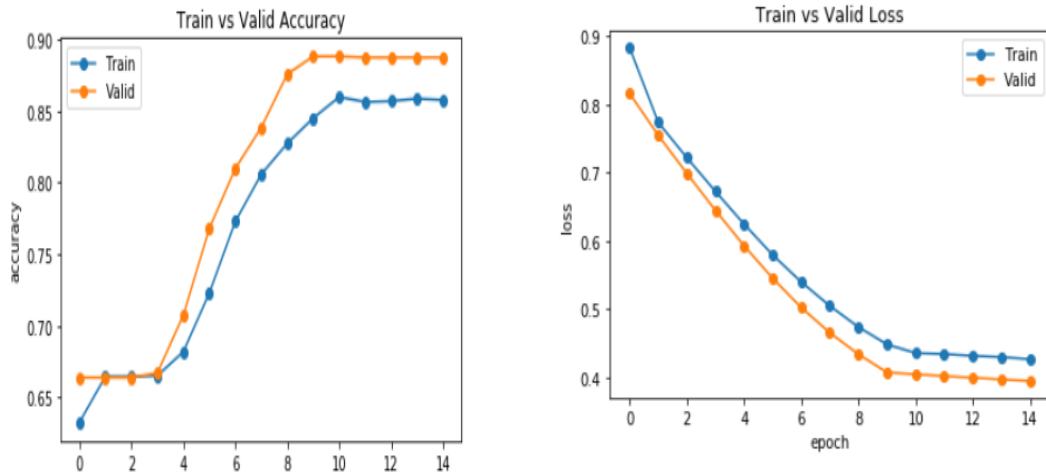
**Figure 5.7:** Comparison plot for Quantum Filter vs Classical Filter in exp.1



**Figure 5.8:** Classification Metrics For Model 1.1,1.2 and 1.3

### 5.2.2 Experiment No.2

In the second experiment we kept the image size same as experiment 1 and use another 3\*3 filter. The input image of size 28\*28 was fit into the classical, trainable and non trainable quantum filter respectively. 13\*13 size feature map is obtained after convolution in 9 channels. Then, 13\*13\*9 feature vector is then flattened into 1521 one dimensional feature and passed into fully connected layer and in the output class predicted by softmax activation function.



(a) Training vs Validation accuracy

(b) Training vs Validation loss

**Figure 5.9:** Training vs Validation curve of Model 2.1

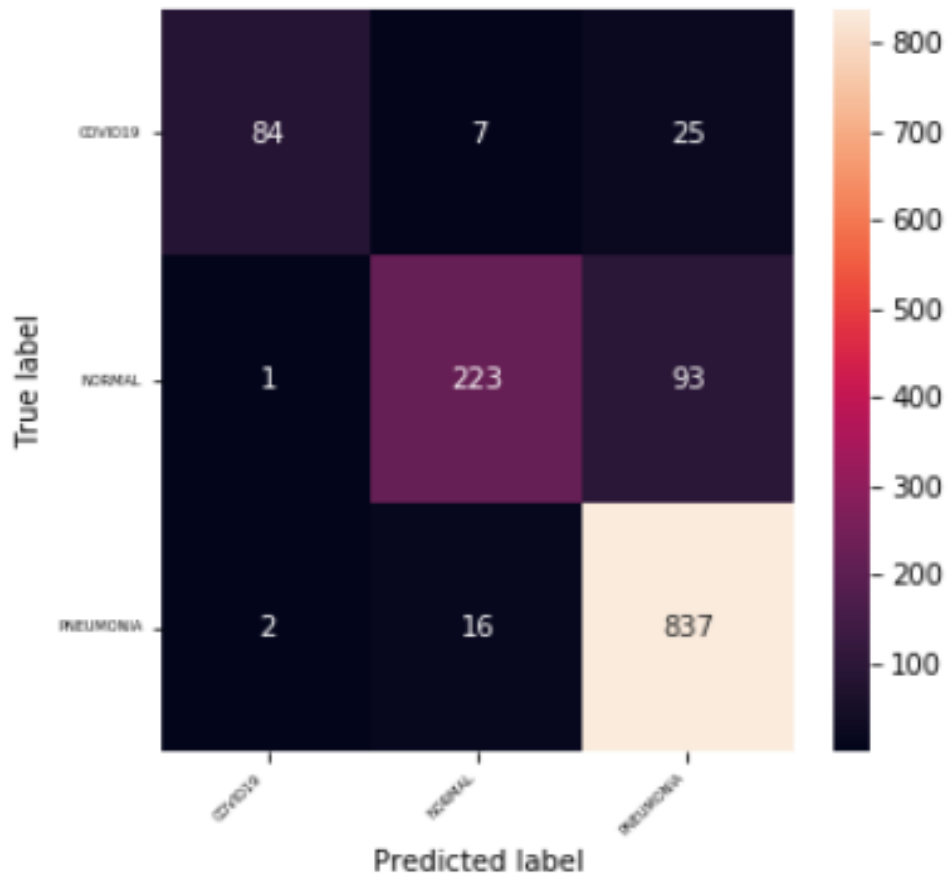


Figure 5.10: Confusion matrix of Model 2.1

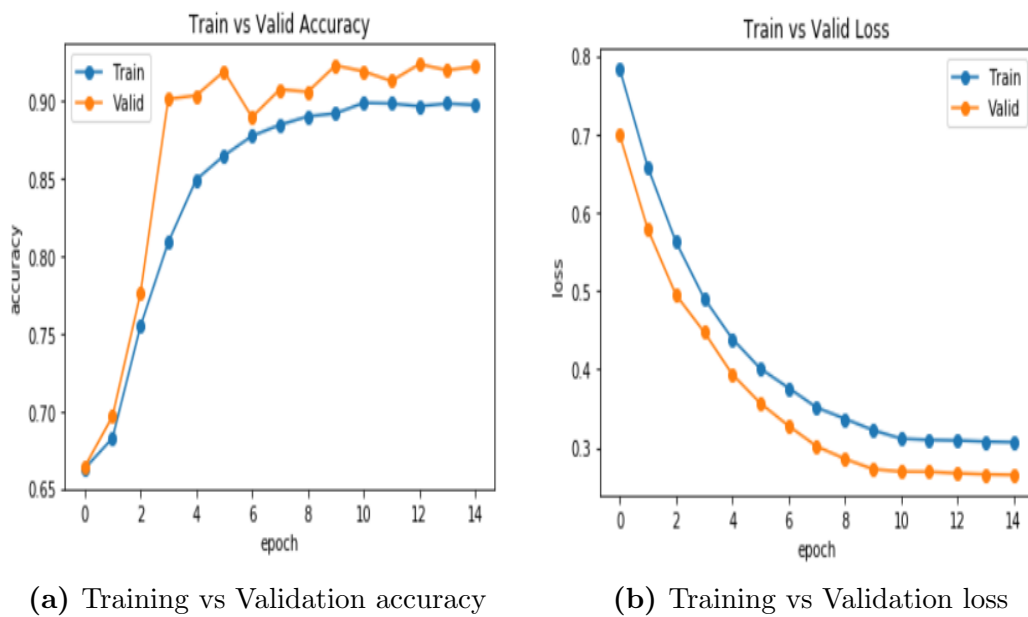


Figure 5.11: Training vs Validation curve of Model 2.2

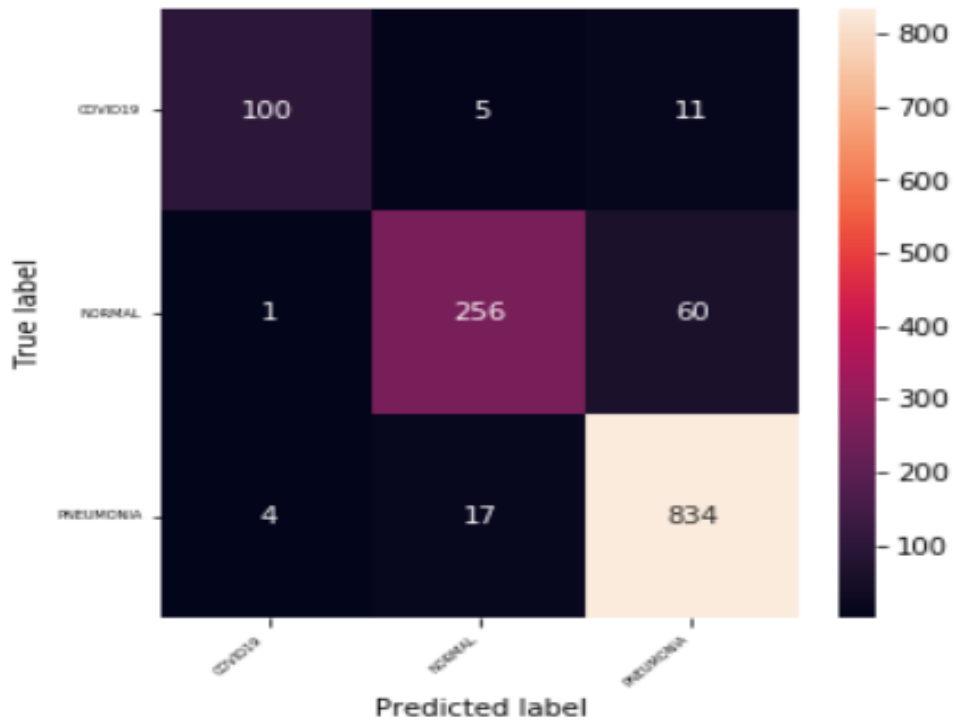
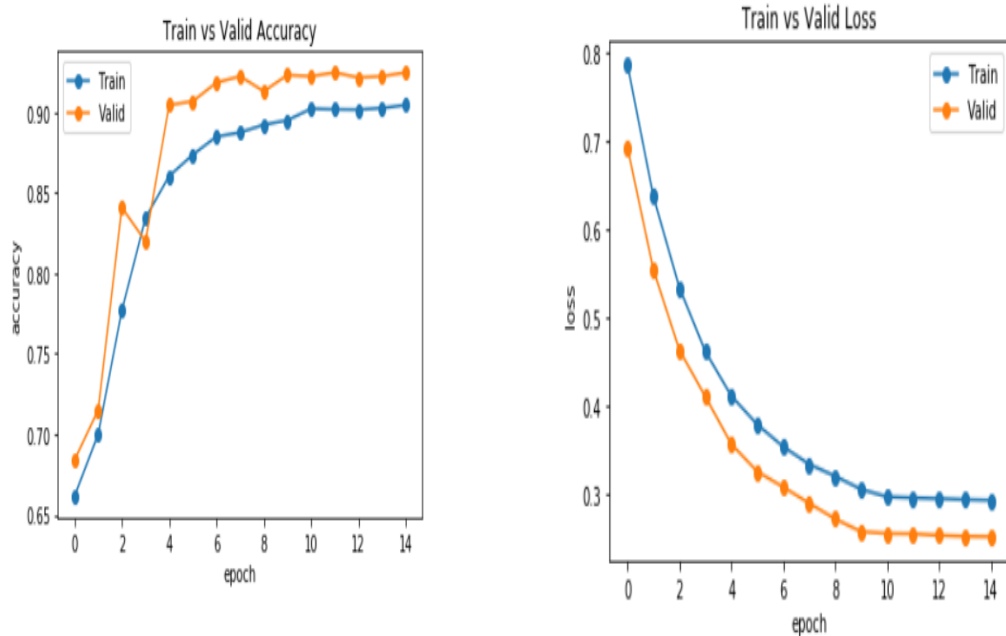


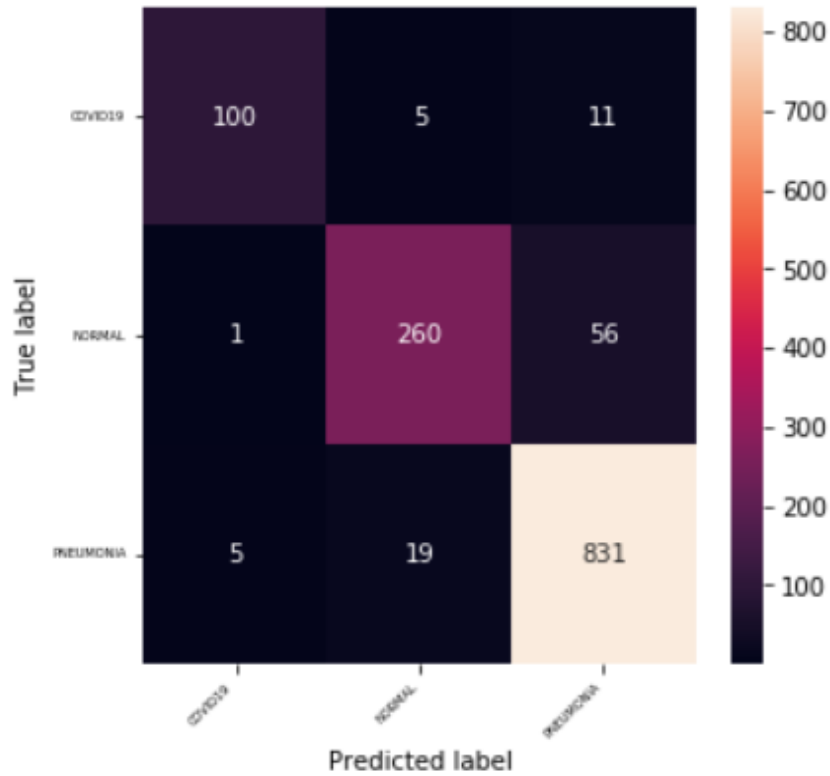
Figure 5.12: Confusion matrix of Model 2.2



(a) Training vs Validation accuracy

(b) Training vs Validation loss

Figure 5.13: Training vs Validation curve of Model 2.3



**Figure 5.14:** Confusion matrix of Model 2.3

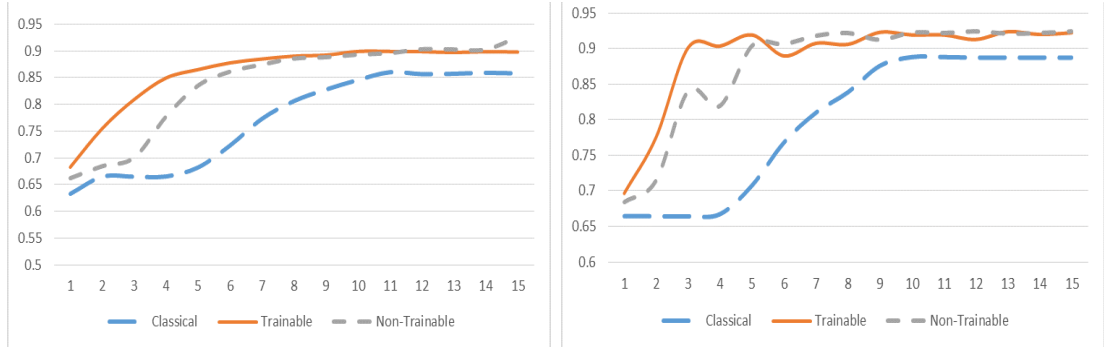
The accuracy curve shows that training accuracy for quantum filter is higher than classical filter. Here in this experiment the learning process in quantum in quantum trainable filter and non-trainable filter was quite different. Both of them converge at 8th epoch and the classical model converged at 11th epoch. The best training accuracy for model 2.1, 2.2, 2.3 were 0.8577, 0.9239, 0.9247 respectively. The validation accuracy curve was quite zig-zag in all three models.

In the case of loss curve the loss of quantum filter was better. But there was no big difference between the non-trainable filter and the trainable filter. The non-trainable and trainable filter have converged at epoch 6 in the loss curve. The classical model converged at epoch 11. The best train loss for model 2.1, 2.2 and 2.3 were 0.4267, 0.3066 and 0.2941 respectively. Similarly the quantum filters have greater performance than the classical in all three models.

The summary of classification of each model are listed below:

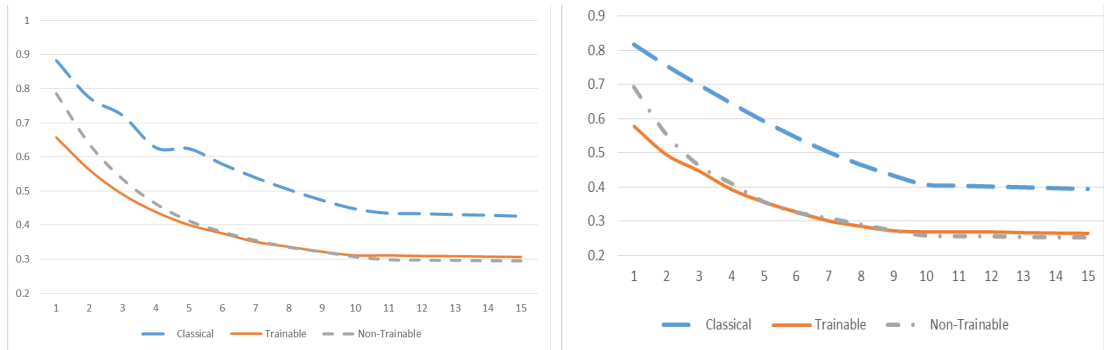
Model No.	Training Time(Min)	No of. Trainable Parameters	Accuracy	Precision	Recall	F1-Score
2.1	58	4656	0.888	0.8	0.92	0.8466
2.2	247	4569	0.9239	0.883	0.93	0.893
2.3	301	4576	0.9247	0.8833	0.93	0.9

**Table 5.5:** Model Summary of Model No 2.1,2.2 and 2.3



**(a)** Training accuracy of Quantum Filter vs Classical Filter

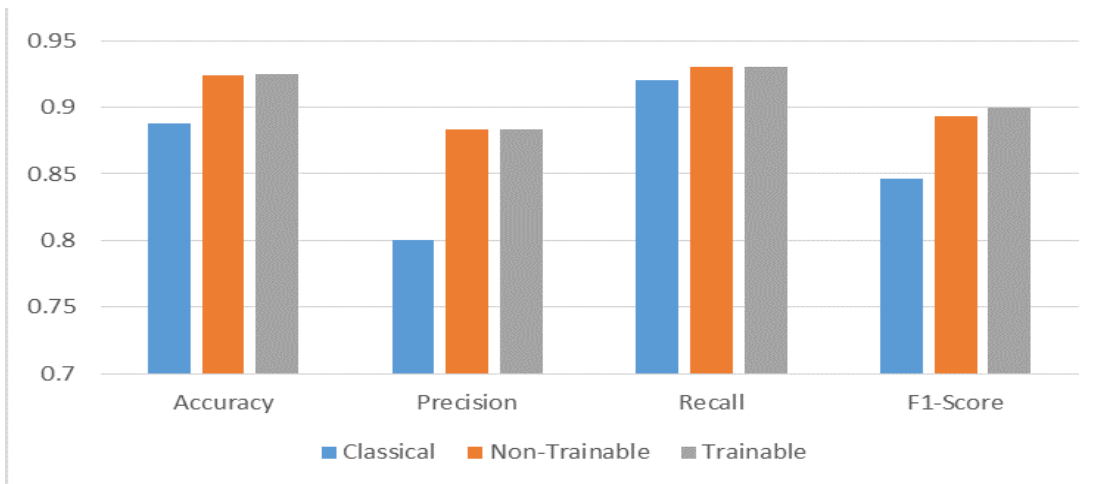
**(b)** Validation accuracy of Quantum Filter vs Classical Filter



**(c)** Training loss of Quantum Filter vs Classical Filter

**(d)** Validation loss of Quantum Filter vs Classical Filter

**Figure 5.15:** Comparison plot for Quantum Filter vs Classical Filter in exp.2



**Figure 5.16:** Classification Metrics For Model 2.1,2.2 and 2.3



### 5.2.3 Experiment No.3

In experiment no .3 we take 4 by 4 filter to map the input image features into sixteen channels. But, the image dimension was same 28\*28. The grayscale input image is convoluted by classical filter and then by the quantum trainable filter to obtain 13\*13\*16 feature map. The 13\*13\*16 feature map is flattened in 2704 one dimensional feature vector. Then predicted by the softmax activation function in the output layer.

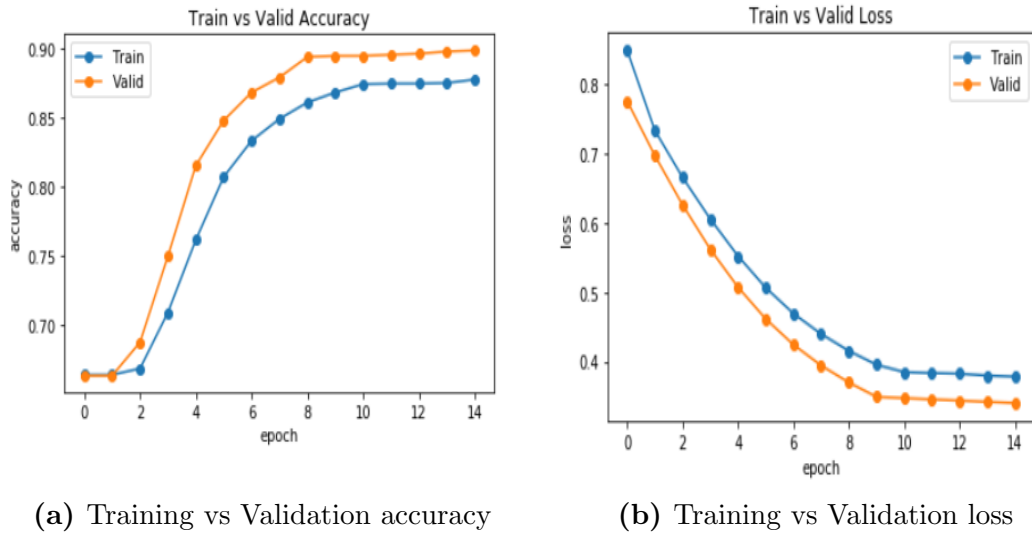


Figure 5.17: Training vs Validation curve of Model 3.1

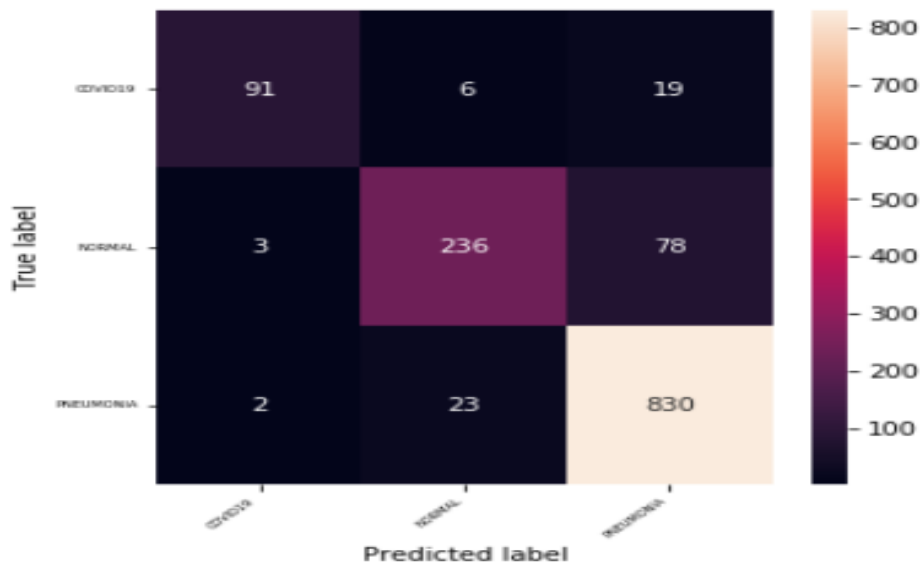
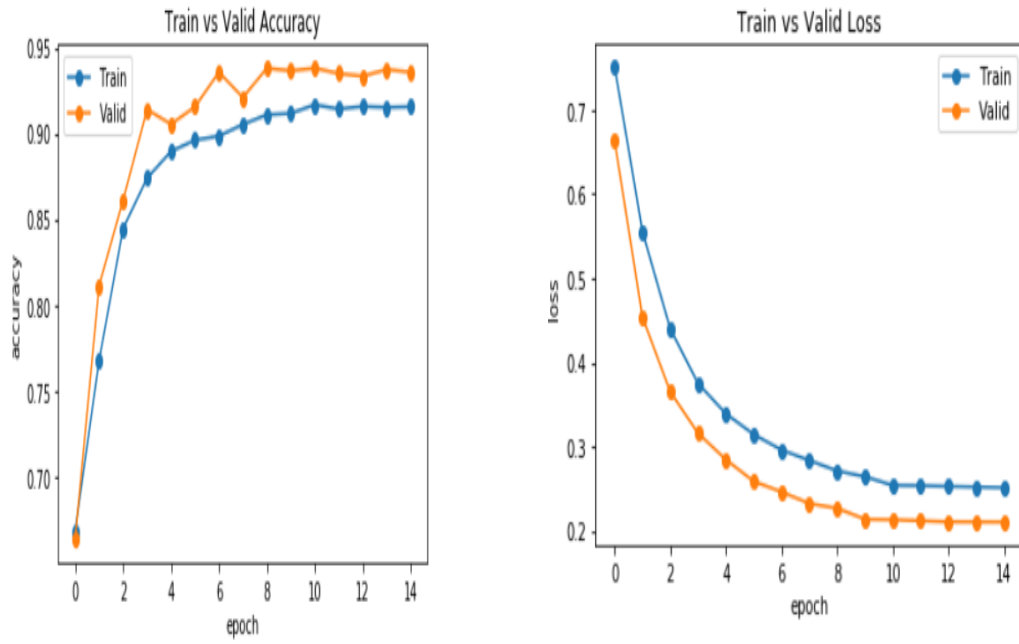


Figure 5.18: Confusion matrix of Model 3.1



(a) Training vs Validation accuracy

(b) Training vs Validation loss

Figure 5.19: Training vs Validation curve of Model 3.2

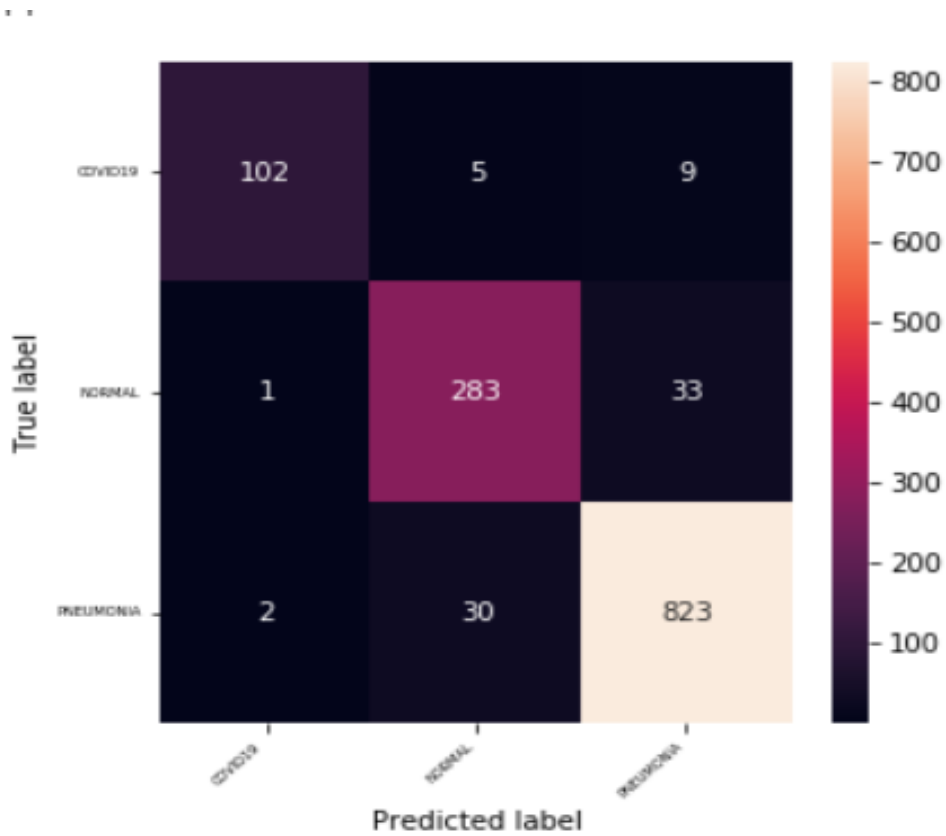
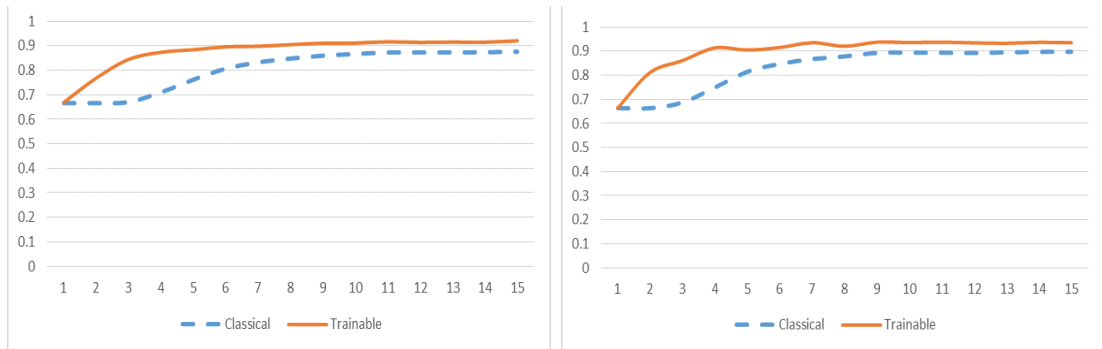


Figure 5.20: Confusion matrix of Model 3.2

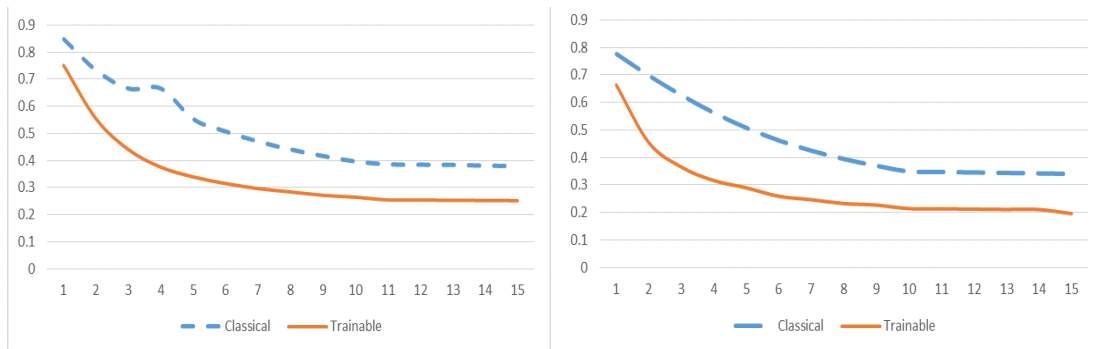
The quantum trainable filter model started to train exponentially from epoch 1 to epoch 4 and then converged. The classical convolution model slowly started learning from the training set and converged from epoch 7. Similarly, in the loss curve the quantum filter loss was stable and converged from 3rd epoch whereas loss was zig-zak for the classical filter. The accuracy of quantum filter was much better than the classical filter as quantum filter obtained the feature map in 16 channels, whereas in the classical filter large sized even filter could not extract symmetric features so the accuracy was low. The summary of classification of each model are listed below:

Model No.	Training Time(min)	No. of. Trainable Parameters	Accuracy	Precision	Recall	F1-Score
3.1	84	8387	0.8096	0.8066	0.8466	0.82
3.2	1282	8126	0.9379	0.8833	0.9366	0.9233

**Table 5.6:** Model Summary of Model No 3.1 and 3.2

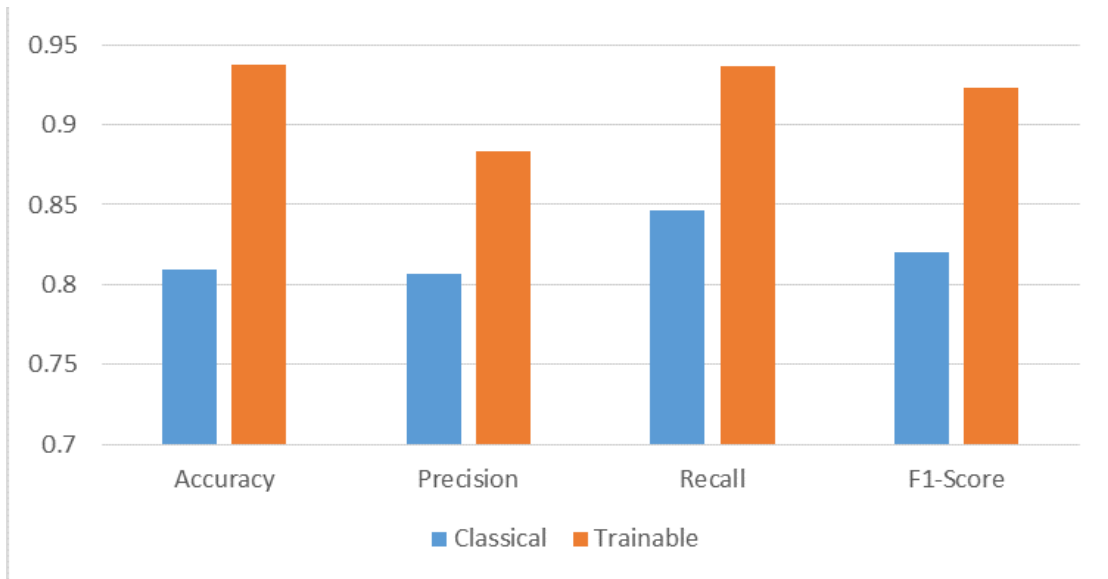


(a) Training accuracy of Quantum Filter vs Classical Filter (b) Validation accuracy of Quantum Filter vs Classical Filter



(c) Training loss of Quantum Filter vs Classical Filter (d) Validation loss of Quantum Filter vs Classical Filter

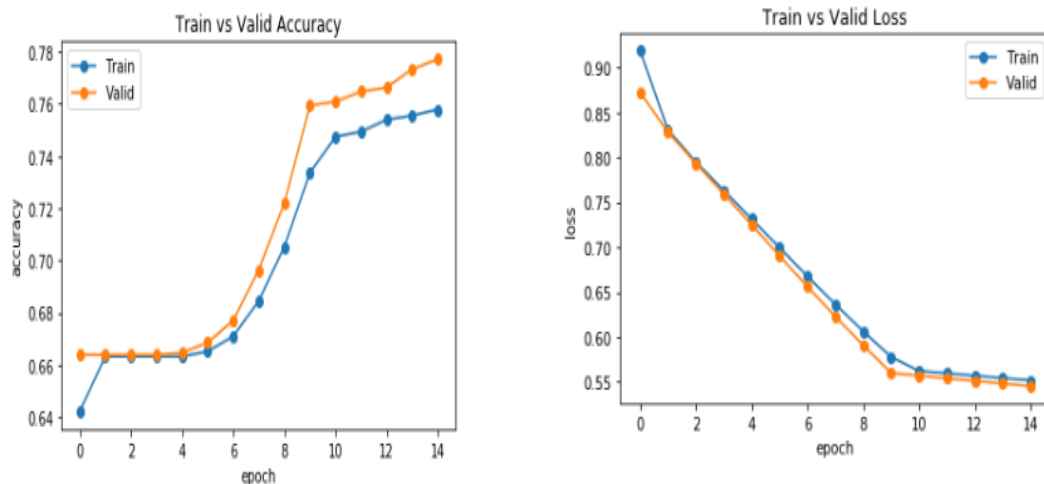
**Figure 5.21:** Comparison plot for Quantum Filter vs Classical Filter in exp.3



**Figure 5.22:** Classification Metrics For Model 3.1 and 3.2

#### 5.2.4 Experiment no.4

The dimension of the image was increased from  $28 \times 28$  to  $32 \times 32$  in this experiment. First the input  $32 \times 32$  size gray scale image is passed to the classical filter and then quantum filter to obtain  $16 \times 16$  features map in the four channels. Then  $16 \times 16 \times 4$  features vector is flattened into 1024 and passed to the fully connected layer which has 3 output nodes. In the last layer of the model there was software activation function.



**(a)** Training vs Validation accuracy      **(b)** Training vs Validation loss

**Figure 5.23:** Training vs Validation curve of Model 4.1

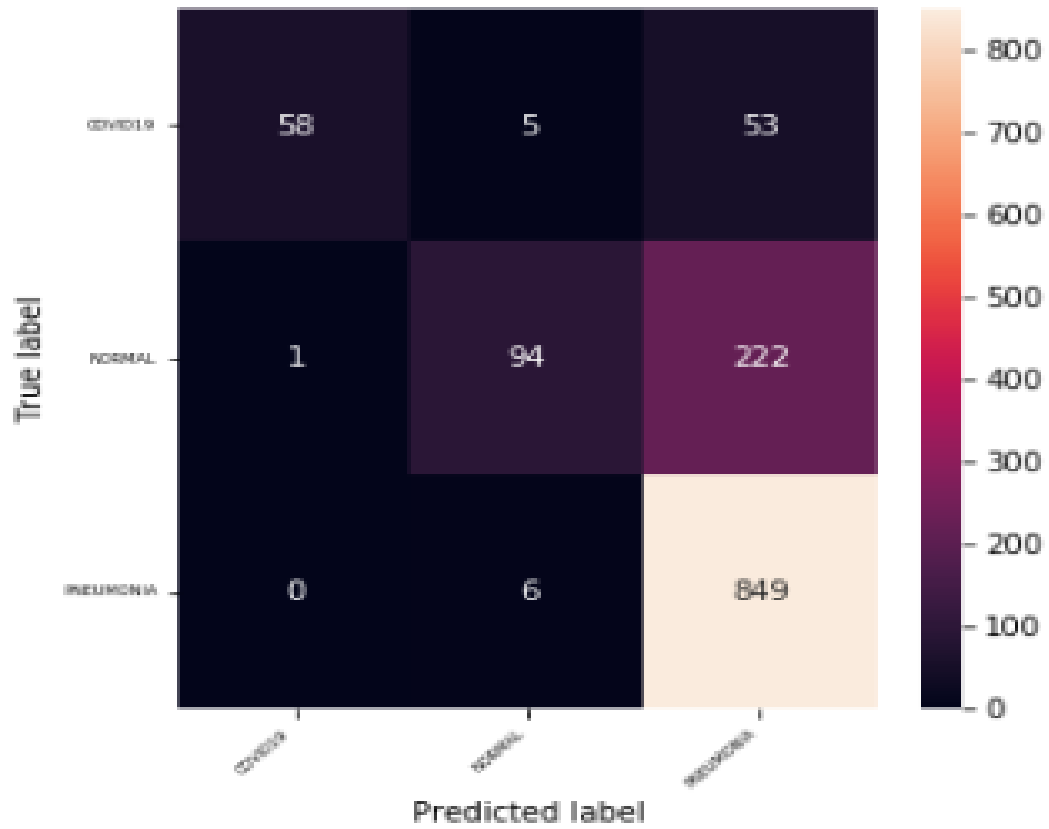
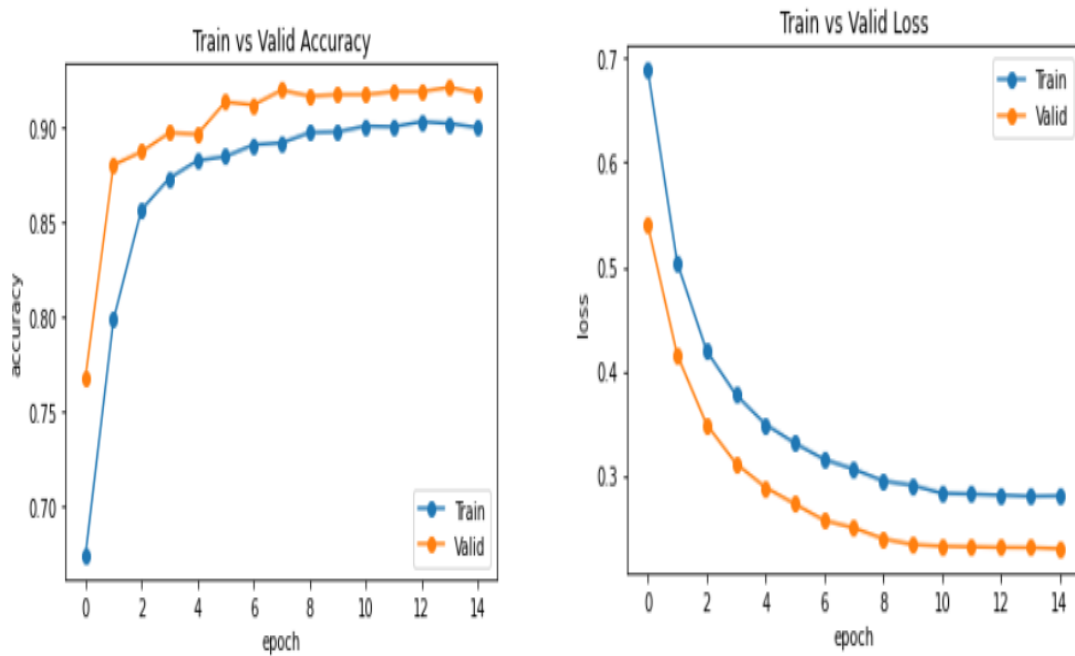


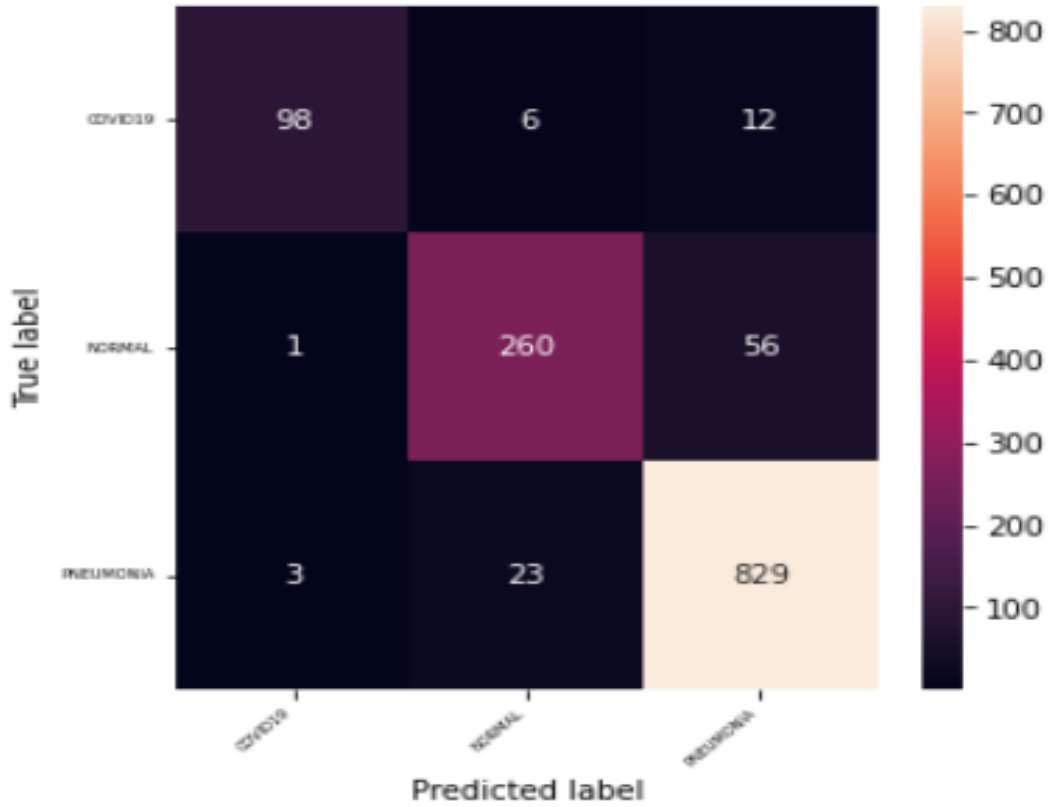
Figure 5.24: Confusion matrix of Model 4.1



(a) Training vs Validation accuracy

(b) Training vs Validation loss

Figure 5.25: Training vs Validation curve of Model 4.2



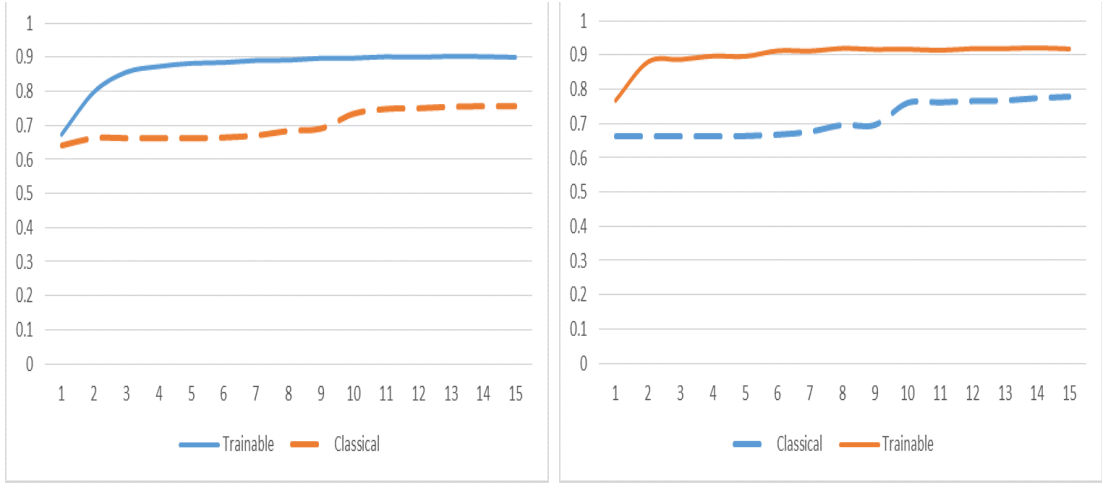
**Figure 5.26:** Confusion matrix of Model 4.2

Similar to the above experiment the training of quantum filter was more stable than of classical filter. The quantum filter converged at epoch 4 where as the classical filter only converged at epoch 10. The quantum filter starts to learn exponentially after epoch 0 but the classical filter start learning properly after epoch 3 and 4. The best train accuracy for the classical convolution was 0.8225 and for the quantum filter was 0.9003. The best train loss for classical model was 0.4866 and for the quantum filter was 0.2801. The accuracy of the classical filter has increased to 28\*28 size as it could extract more features from higher dimensional images. As compared to the quantum filter the classical filter accuracy was quite low.

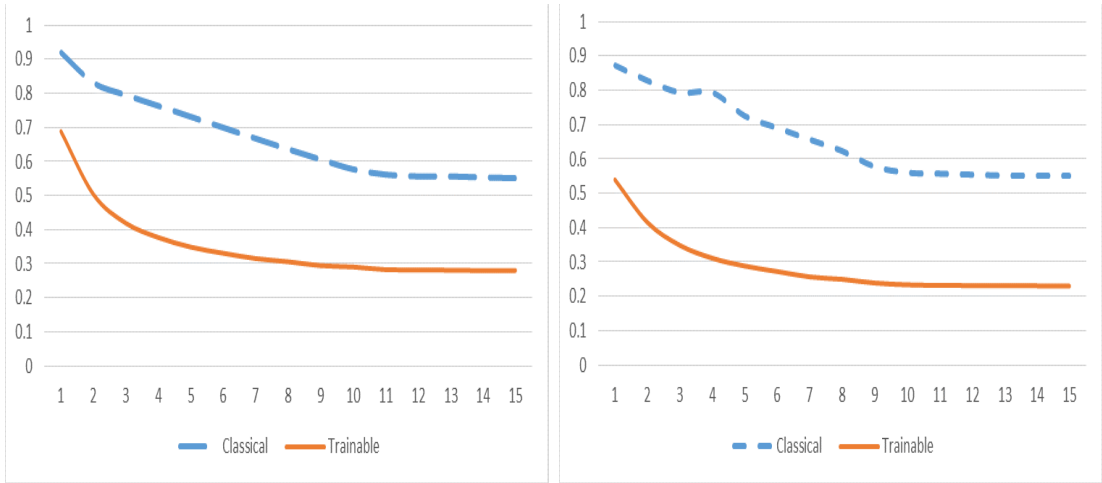
The summary of classification of each model are listed below:

Model No.	Training Time(min)	No of. Trainable Parameters	Accuracy	Precision	Recall	F1-Score
4.1	37	3102	0.7772	0.59	0.88	0.6566
4.2	296	3082	0.9215	0.8766	0.9266	0.9033

**Table 5.7:** Model Summary of Model No 4.1 and 4.2

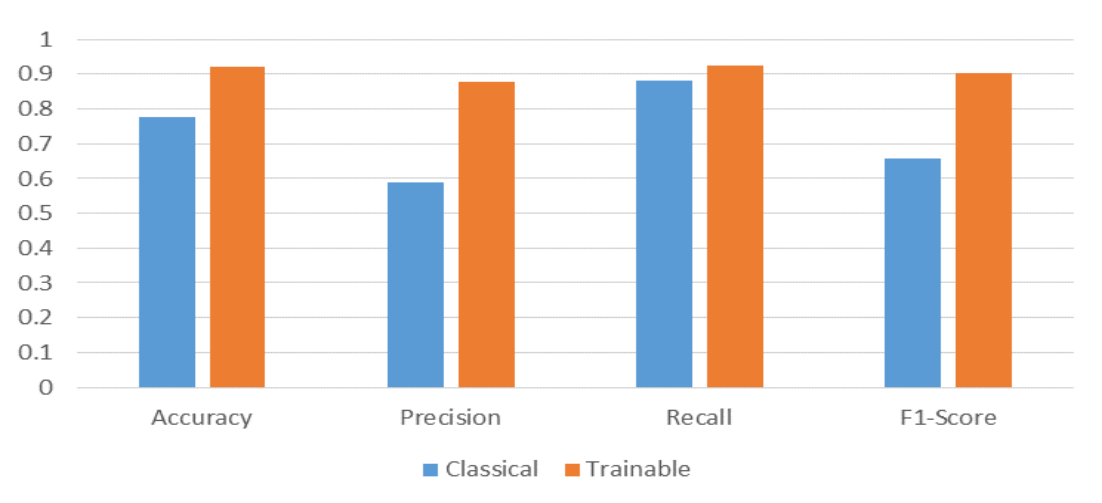


(a) Training accuracy of quantum filter vs classical filter (b) Validation accuracy of quantum filter vs classical filter



(c) Training loss of quantum filter vs classical filter (d) Validation loss of quantum filter vs classical filter

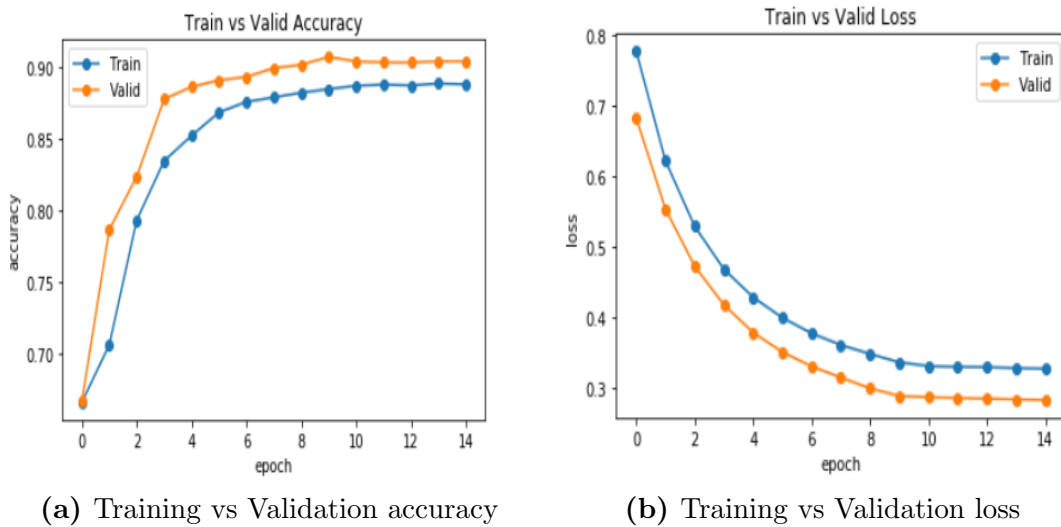
**Figure 5.27:** Comparison plot for Quantum Filter vs Classical Filter in exp.4



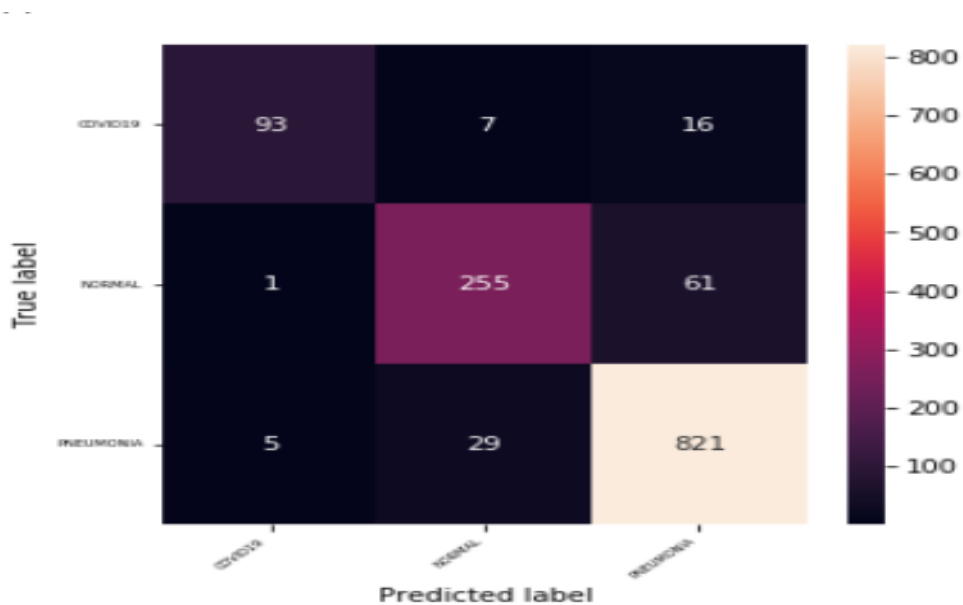
**Figure 5.28:** Classification Metrics For Model 4.1 and 4.2

### 5.2.5 Experiment No 5

The last experiment was performed for higher dimensional image  $64 \times 64$ . The filter size was taken as 2 by 2. The input image of size  $64 \times 64$  is feed into the quantum convolution model and classical convolution model to obtain  $32 \times 32$  size features vectors in 4 channels. The  $32 \times 32 \times 4$  feature vector is than flattened into 4096 one dimensional feature vector and passed to fully connected layer. At last soft-max activation function is used for classification.

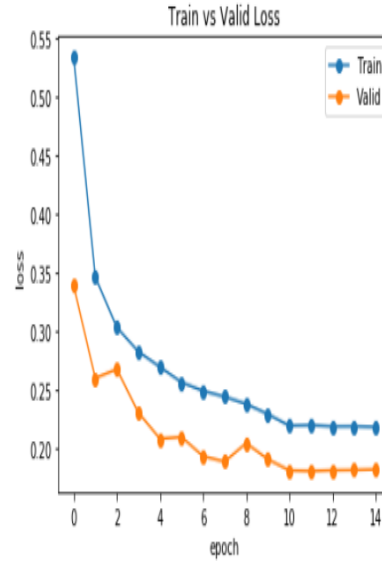
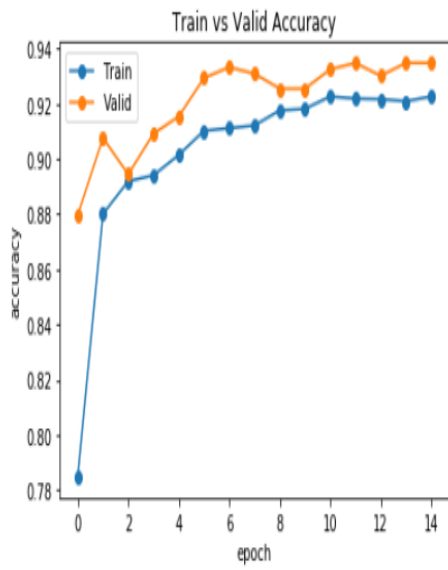


**Figure 5.29:** Training vs Validation curve of Model 5.1



**Figure 5.30:** Confusion matrix of Model 5.1

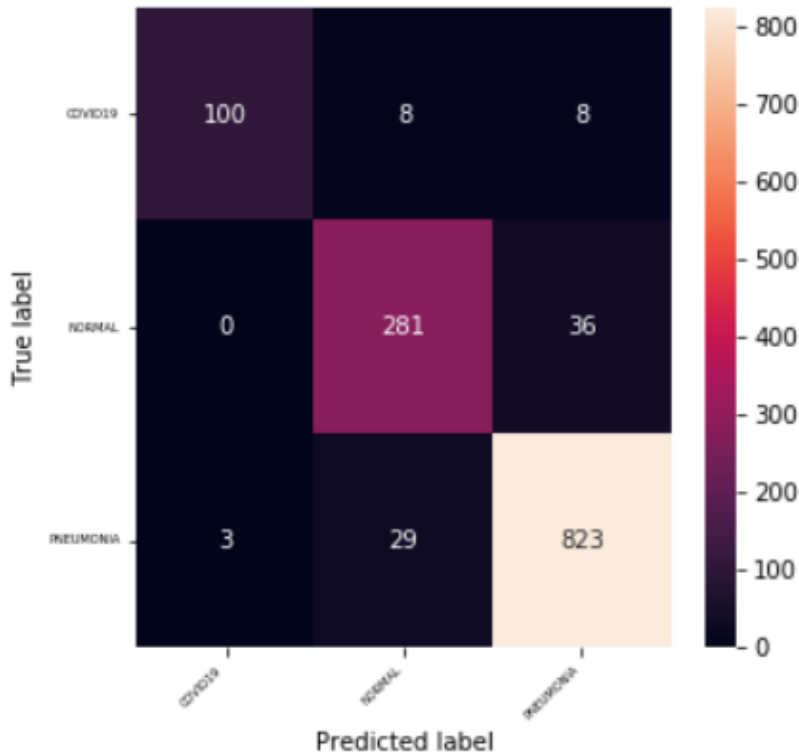




(a) Training vs Validation accuracy

(b) Training vs Validation loss

**Figure 5.31:** Training vs Validation curve of Model 5.2



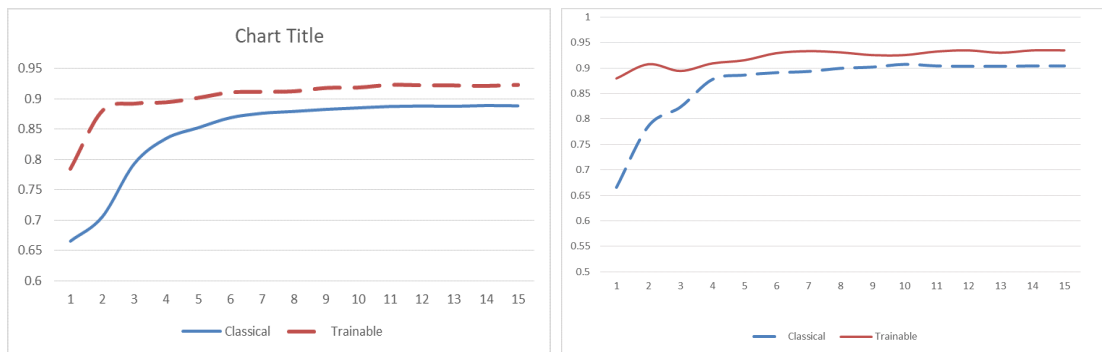
**Figure 5.32:** Confusion matrix of Model 5.2

Similar to other experiment the quantum filter started to learn exponentially from the starting epoch and converged at 5th epoch where as classical filter also started to train from epoch 0 due to increase of size of image and converged at epoch 7. The accuracy and loss of the classical model was improved significantly in this model and was comparable to quantum filter. The best train accuracy for classical and quantum model was 0.884 and 0.9226 respectively. The best train loss for classical and quantum model was 0.3286 and 0.2176.

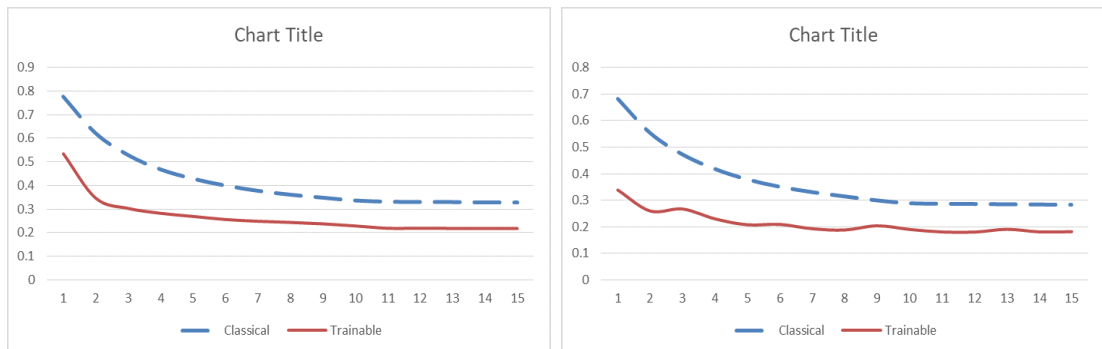
The summary of classification of each model are listed below:

Model No.	Training Time(min)	No of. Trainable Parameters	Accuracy	Precision	Recall	F1-Score
5.1	51	12318	0.9076	0.9	0.9366	0.92
5.2	1157	12298	0.9378	0.85	0.91	0.8833

**Table 5.8:** Model Summary of Model No 5.1 and 5.2

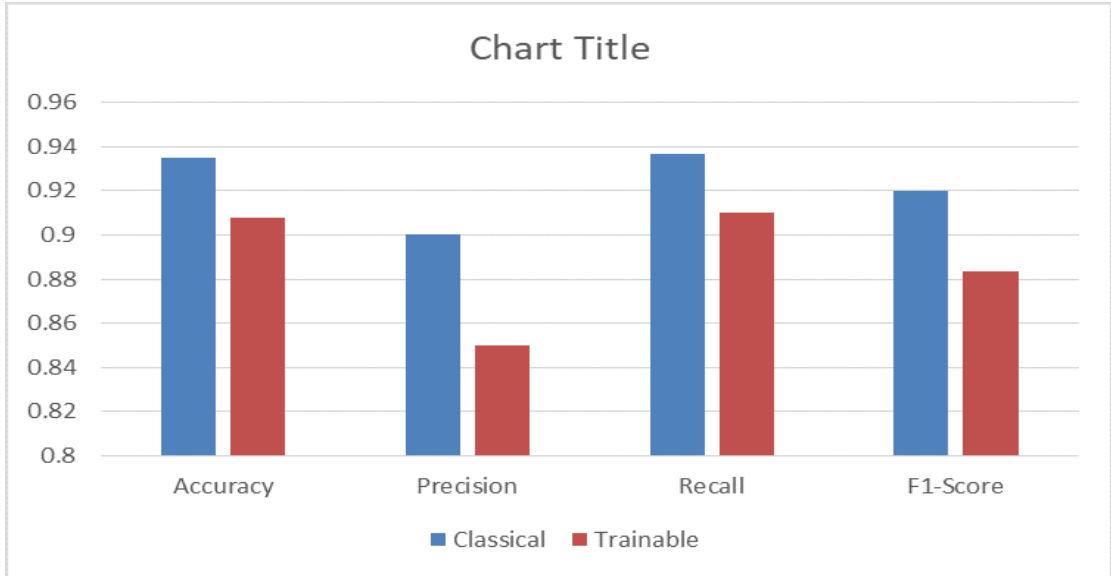


(a) Training accuracy of quantum filter vs classical filter (b) Validation accuracy of quantum filter vs classical filter



(c) Training loss of quantum filter vs classical filter (d) Validation loss of quantum filter vs classical filter

**Figure 5.33:** Comparison plot for Quantum Filter vs Classical Filter in exp.5



**Figure 5.34:** Classification Metrics For Model 5.1 and 5.2

### 5.3 Comparison among Experiments

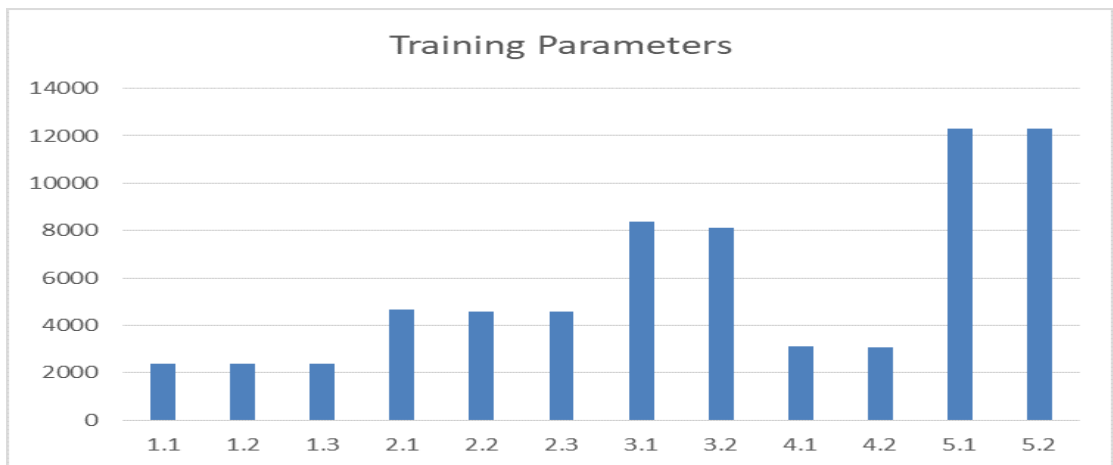
Among all the performed experiment we could gain insight that quantum filter was able to produce significant accuracy and performance in each experiment. The learn-able parameters of quantum filter was less than of classical. Non-trainable quantum filter produced similar accuracy as the trainable with fewer number of parameters according to the gradient, so performance of the quantum filter was also good. The training time of classical model was less as comparable to the quantum model. The quantum model had to be simulated by the local simulator so it increased the training time. Simulating quantum computer in classical increases the complexity exponentially. As the size of filter and size of image increases in both case the training time and number of parameters were also increased. In classical filter case odd filter of 3\*3 performed well as compared to large or small size even filters. But in case of quantum filter increasing the size of filter didn't affect the performance in case of both even and odd filter.

Model No.	Training Parameters	Training Time(Min)	Accuracy
1.1	2375	36	0.7931
1.2	2355	142	0.9239
1.3	2358	212	0.9281
2.1	4656	58	0.888
2.2	4569	247	0.9239
2.3	4576	301	0.9247
3.1	8387	84	0.8096
3.2	8126	1282	0.9379
4.1	3102	37	0.7772
4.2	3082	296	0.9215
5.1	12318	51	0.9076
5.2	12298	1157	0.9348

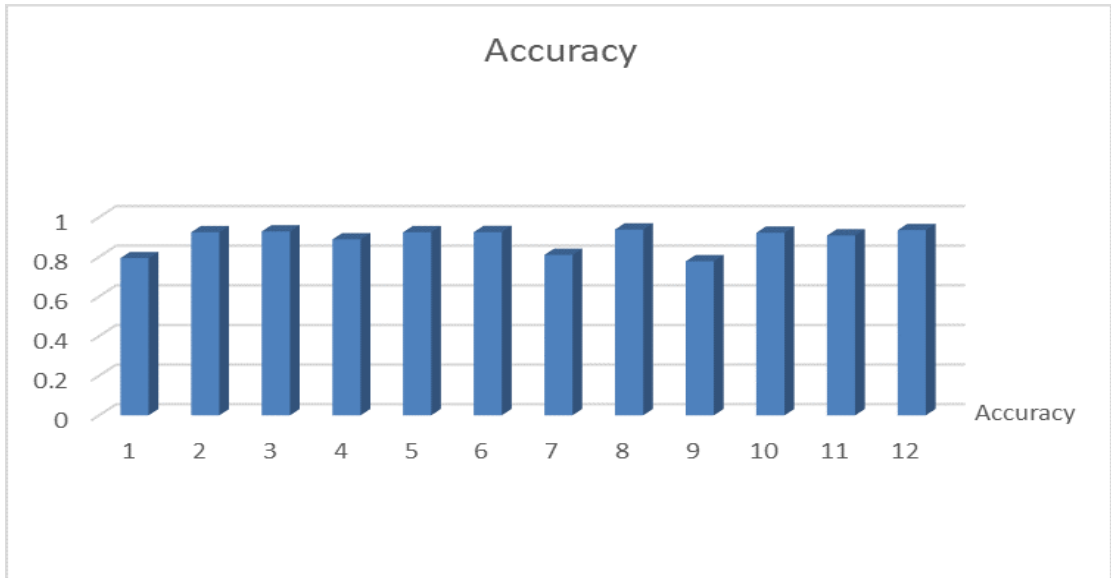
**Table 5.9:** Results of all models



**Figure 5.35:** Training Time of all models



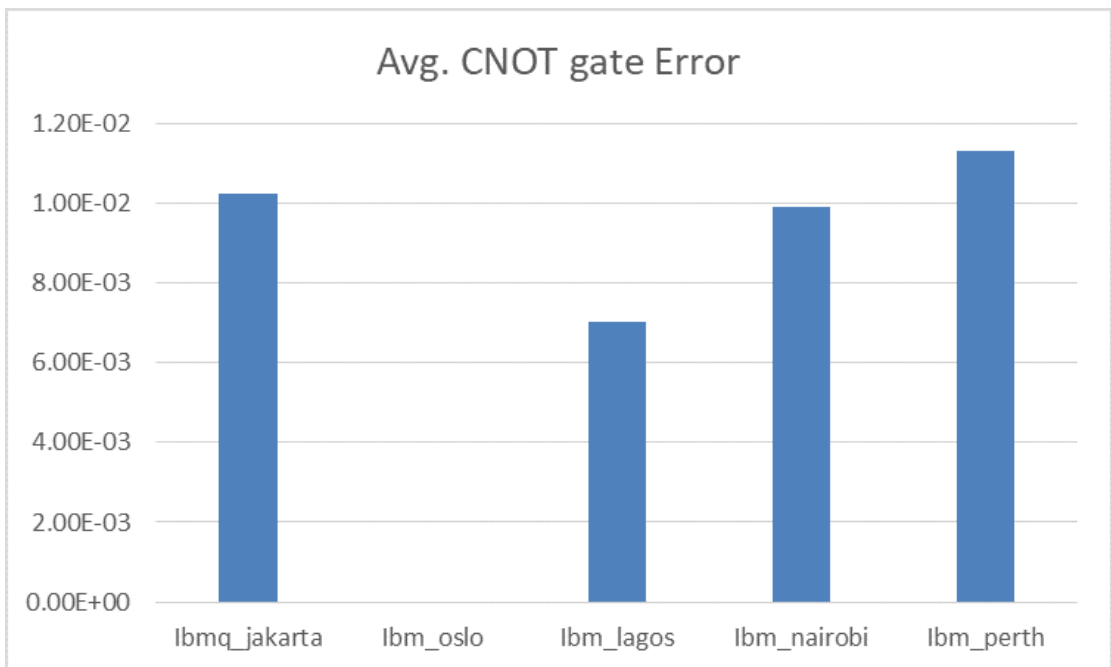
**Figure 5.36:** Training Parameters of all models



**Figure 5.37:** Accuracy of all models

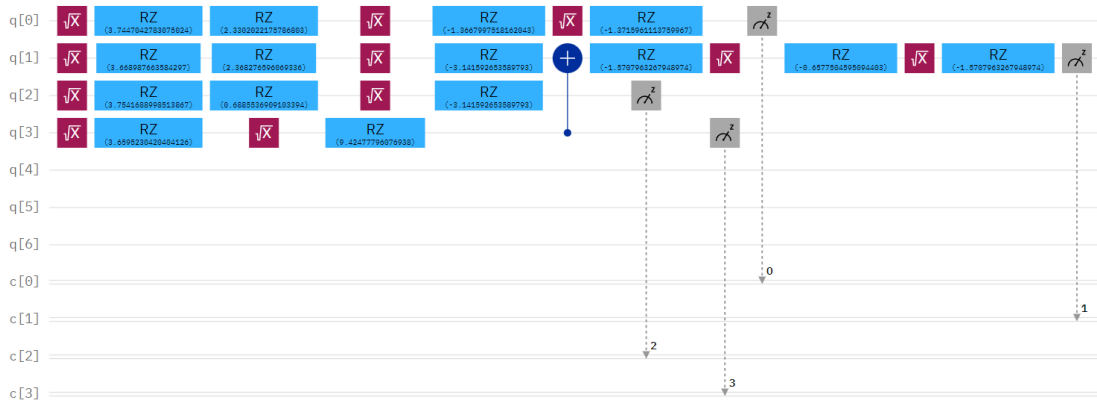
#### 5.4 Model Evaluation in Real Quantum Devices

Real Quantum devices are quite noisy than the simulators, increasing the number of qubits and C-NOT gates in the quantum circuit increases the error in measurement in quantum circuit. The noise rates for different quantum devices are different. The average distribution of error of C-NOT graph of each quantum devices are shown by graph below: The model that was used for testing was quantum convolutional

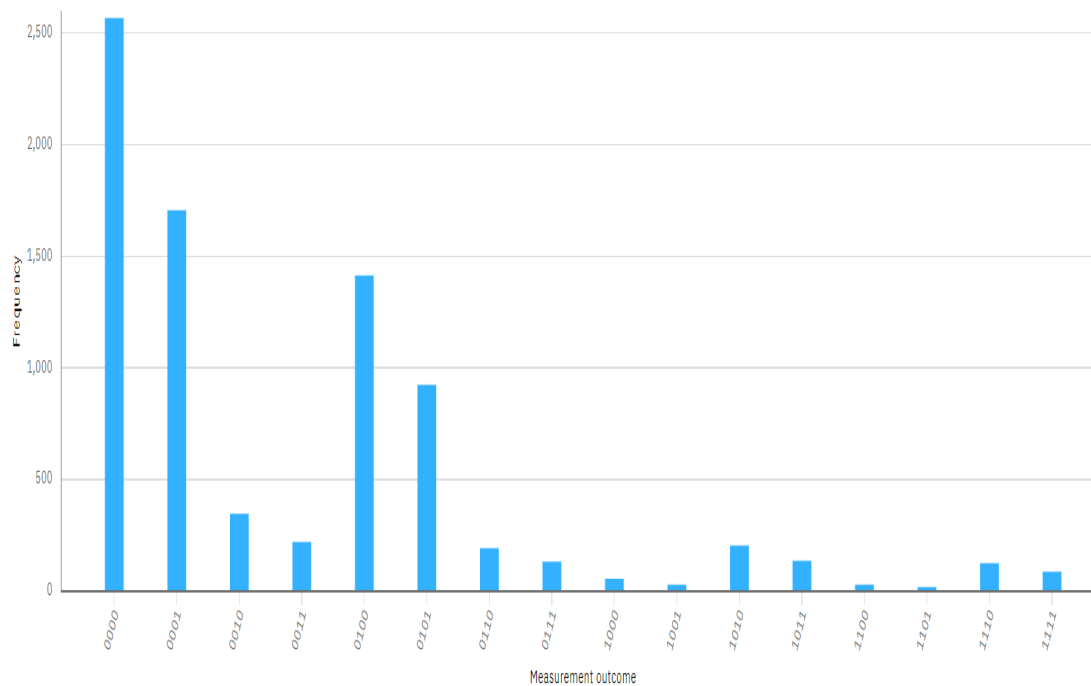


**Figure 5.38:** Avg. CNOT gate error in real quantum devices

model with 28\*28 size input images with 4 qubits. This quantum circuit used only one CNOT-gate so there would be less error in real quantum device too. The job in real quantum devices was sent from Google Colab API. There were 8192 shots per each job and number of circuit per jobs were 10. The run time for each job was different for different jobs and quantum devices.



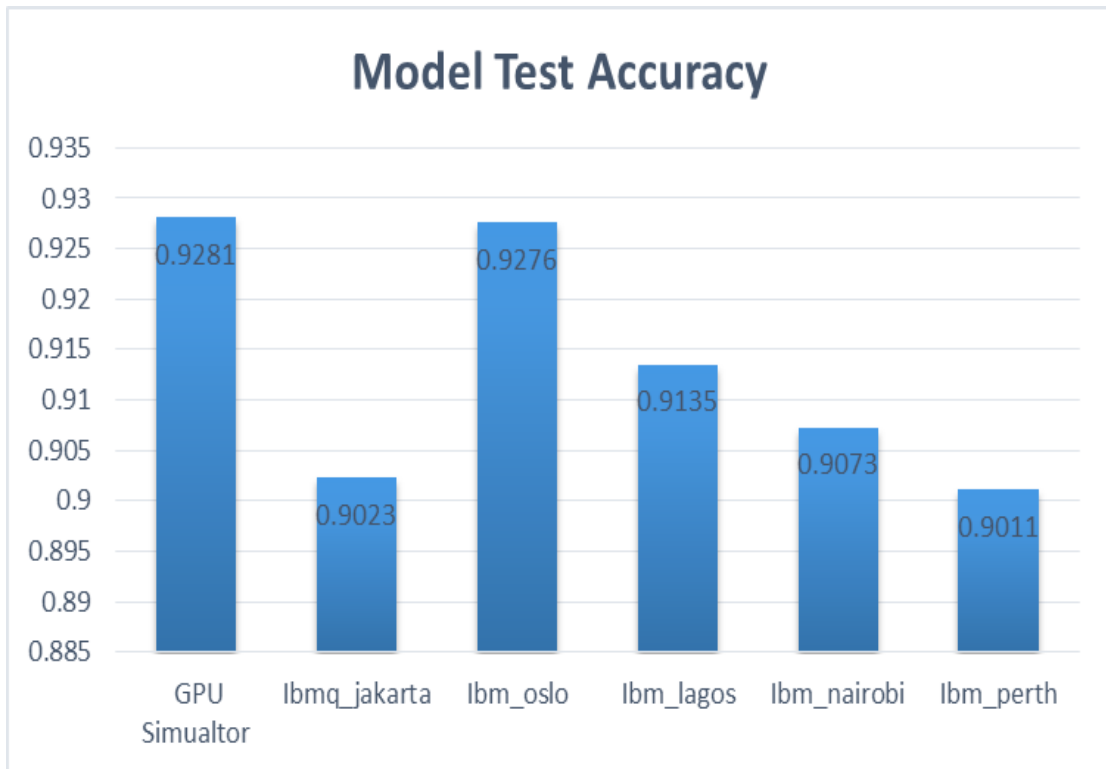
**Figure 5.39:** Instance of quantum circuit after transpiling



**Figure 5.40:** Instance of quantum circuit showing frequency plot of each state

The models were tested in 1288 test set images and different accuracy were found

in different devices shown as below:



**Figure 5.41:** Histogram Showing Model Accuracy in different real quantum devices

The results of model performance on different devices used based on the CNOT gate errors. Real quantum devices with more CNOT gate error had slight less accuracy than the other one. The accuracy of the model was highest in noise free local simulator. On the real quantum devices the accuracy was best for ibmq-oslo and lowest was for the ibmq-perth.

## CHAPTER 6

### CONCLUSION AND FUTURE ENHANCEMENTS

#### 6.1 Conclusion

This research work proposed a hybrid deep learning model for image classification. This work shows a novel quantum convolution approach that is analogous to the classical convolution layer. The experiment performed on this research are based on  $28 \times 28$ ,  $32 \times 32$  and  $64 \times 64$  size input chest x-ray images for  $2 \times 2$ ,  $3 \times 3$  and  $4 \times 4$  quantum convolution filter.

There may be many pandemic that may arise in nearby future such as pandemic could affect the parts parts that supports medical imaging. To outbreak the chain of the pandemic we must identify the victim and isolate them. As the medical data may be in few amount due to their expensiveness and in securities. So, we should be able to build a classification model with less training set with high confidence and accuracy.

So, the model would assist the doctor and other professional. Also CNN as being the best feature extractor among all the machine learning models, but it require a lot training complexity and time complexity. So, we have think about light weight efficient model for image classification. The best model for classical computing with  $2 \times 2$  filter was 0.7931. This is quite low for image classification model as it would require high precision and accuracy for detecting the disease. Introduction of quantum convolution layer in replacement of classical convolution with less amount of trainable parameters produced accuracy 0.9239. The best classical convolution model for this work was  $64 \times 64$  size images with 1213 learnable parameters. The best hybrid quantum classical model was  $64 \times 64$  size input image and  $2 \times 2$  quantum filter having accuracy of 0.9348 with 12298 trainable parameters. The most computationally efficient model had 2355 trainable parameters with 0.9239. Similarly, in each performed experiment trainable filter achieved greater performance than non-trainable filter. However due to small size quantum filter the training process was very high.



This research work shows that in small training datasets hybrid quantum convolution model outperforms classical convolution model in terms of accuracy no. of trainable parameters.

## 6.2 Limitations and Future enhancement

This experiment was performed mostly on low resolution images,would be performed other experiment of higher resolution images.This research works only replace a single convolution layer,other classical convolution layer in the hybrid model could also be replaced.Hyperparameters in this research work was mostly kept constant ,few more hyperparameters could be tuned.Three quantum circuit was only used throughout this work,other various quantum circuit may be also be used for feature extraction.Being NISQ era avaiability of less number of noiseless qubits.So,in this work we perform quantum convolution strips by strips training this kind of hybrid quantum classical model is time consuming process.

The IBM quantum Researchers program only provided 180 reservtaion time for five backends so we could run a classification model in more real quantum devices.Further more the enhancement of this works may be:

- Training the entire model in real quantum devices on more simulators and real quantum devices using parameter shifting rules.
- Improving the classification model performance by introducing few more image datasets in COVID class.
- Use of higher dimenisonal quantum filter for quantum convolution.
- Deployment of model in real world applications.

## Bibliography

- [1] Wei-jie Guan, Zheng-yi Ni, Yu Hu, Wen-hua Liang, Chun-quan Ou, Jian-xing He, Lei Liu, Hong Shan, Chun-liang Lei, David SC Hui, et al. Clinical characteristics of coronavirus disease 2019 in china. *New England journal of medicine*, 382(18):1708–1720, 2020.
- [2] V Corman, T Bleicker, S Brünink, C Drosten, O Landt, and M Koopmans. Zambon public health england, m. diagnostic detection of 2019-ncov by real-time rt-rer incl. *Euro Surveill*, 25:2000045, 2020.
- [3] Matthew Hayward. Quantum computing and shor’s algorithm. *Sydney: Macquarie University Mathematics Department*, 2008.
- [4] Christof Zalka. Grover’s quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746, 1999.
- [5] Amer Delilbasic, Gabriele Cavallaro, Madita Willsch, Farid Melgani, Morris Riedel, and Kristel Michielsen. Quantum support vector machine algorithms for remote sensing data classification. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, pages 2608–2611. IEEE, 2021.
- [6] M Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J Coles. Variational quantum state eigensolver. *arXiv preprint arXiv:2004.01372*, 2020.
- [7] Andrea Mari, Thomas R Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4:340, 2020.
- [8] Harshit Mogalapalli, Mahesh Abburi, B Nithya, and Surya Kiran Vamsi Bandreddi. Classical–quantum transfer learning for image classification. *SN Computer Science*, 3(1):1–8, 2022.
- [9] Muhammad Junaid Umer, Javeria Amin, Muhammad Sharif, Muhammad Al-mas Anjum, Faisal Azam, and Jamal Hussain Shah. An integrated framework

- for covid-19 classification based on classical and quantum transfer learning from a chest radiograph. *Concurrency and Computation: Practice and Experience*, 34(20):e6434, 2022.
- [10] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.
- [11] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quconvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2(1):1–9, 2020.
- [12] Prabhjot Kaur, Shilpi Harnal, Rajeev Tiwari, Fahd S Alharithi, Ahmed H Almulihi, Irene Delgado Noya, and Nitin Goyal. A hybrid convolutional neural network model for diagnosis of covid-19 using chest x-ray images. *International Journal of Environmental Research and Public Health*, 18(22):12191, 2021.
- [13] Emtiaz Hussain, Mahmudul Hasan, Md Anisur Rahman, Ickjai Lee, Tasmi Tamanna, and Mohammad Zavid Parvez. Corodet: A deep learning based classification for covid-19 detection using chest x-ray images. *Chaos, Solitons & Fractals*, 142:110495, 2021.
- [14] Rubina Sarki, Khandakar Ahmed, Hua Wang, Yanchun Zhang, and Kate Wang. Automated detection of covid-19 through convolutional neural network using chest x-ray images. *Plos one*, 17(1):e0262052, 2022.
- [15] Essam H Houssein, Zainab Abohashima, Mohamed Elhoseny, and Waleed M Mohamed. Hybrid quantum-classical convolutional neural network model for covid-19 prediction using chest x-ray images. *Journal of Computational Design and Engineering*, 9(2):343–363, 2022.
- [16] Kinshuk Sengupta and Praveen Ranjan Srivastava. Quantum algorithm for quicker clinical prognostic analysis: an application and experimental study using ct scan images of covid-19 patients. *BMC Medical Informatics and Decision Making*, 21(1):1–14, 2021.
- [17] Shikhar Johri, Mehendi Goyal, Sahil Jain, Manoj Baranwal, Vinay Kumar, and Rahul Upadhyay. A novel machine learning-based analytical framework

- for automatic detection of covid-19 using chest x-ray images. *International Journal of Imaging Systems and Technology*, 31(3):1105–1119, 2021.
- [18] Ezz El-Din Hemdan, Marwa A Shouman, and Mohamed Esmail Karar. Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*, 2020.
- [19] Xiaoming Li, Wenbing Zeng, Xiang Li, Haonan Chen, Linping Shi, Xinghui Li, Hongnian Xiang, Yang Cao, Hui Chen, Chen Liu, et al. Ct imaging changes of corona virus disease 2019 (covid-19): a multi-center study in southwest china. *Journal of translational medicine*, 18(1):1–8, 2020.
- [20] Md Zabirul Islam, Md Milon Islam, and Amanullah Asraf. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Informatics in medicine unlocked*, 20:100412, 2020.
- [21] Aayush Jaiswal, Neha Gianchandani, Dilbag Singh, Vijay Kumar, and Manjit Kaur. Classification of the covid-19 infected patients using densenet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, 39(15):5682–5689, 2021.
- [22] Mohamed Hibat-Allah, Estelle M Inack, Roeland Wiersema, Roger G Melko, and Juan Carrasquilla. Variational neural annealing. *Nature Machine Intelligence*, 3(11):952–961, 2021.
- [23] Christopher Roth. Iterative retraining of quantum spin models using recurrent neural networks. *arXiv preprint arXiv:2003.06228*, 2020.
- [24] Hanrui Wang, Yongshan Ding, Jiaqi Gu, Zirui Li, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han. Quantumnas: Noise-adaptive search for robust quantum circuits. In *The 28th IEEE International Symposium on High-Performance Computer Architecture (HPCA-28)*, 2022.

## APPENDIX A

### More Results on Quantum Convolution

#### 1. 4 Channel Quantum Convolution

S.N.	Model	Best Train Acc	Best Val Acc	Best Train Loss	Best Val Loss	No.of Trainable Parameters	Training Time
1.1	Quantum Convolution (Non trainable)	0.9024	0.9239	0.2851	0.2416	2355	142m 39s
1.2	Quantum Convolution (8 random parameters)	0.9053	0.9262	0.2737	0.2305	2362	212m 48s
1.3	Quantum Convolution (12 random parameters)	0.9170	0.9269	0.2626	0.2301	2365	267m 9s
1.4	Quantum Convolution (25 random parameters)	0.9156	0.9278	0.2513	0.2124	2374	446m49s

#### 2. 9 Channel Quantum Convolution

S.N.	Model	Best Train Acc	Best Val Acc	Best Train Loss	Best Val Loss	No.of Trainable Parameters	Training Time
2.1	Quantum Convolution (Non Train -8 random parameters)	0.8976	0.9212	0.3091	0.1814	4570	201m40s
2.2	Quantum Convolution (8 random parameters)	0.8985	0.9230	0.3066	0.2649	4573	247m 52s
2.3	Quantum Convolution (12 random parameters)	0.8985	0.9247	0.2941	0.2530	4576	301m 46s
2.4	Quantum Convolution (16 random parameters)	0.9063	0.9216	0.2775	0.2320	4580	353m 51s
2.5	Quantum Convolution (25 random parameters)	0.9077	0.924689	0.2738	0.2349	4586	463m 38s
2.6	Quantum Convolution (35 random parameters)	0.9088	0.9239	0.2553	0.2162	4591	572m 54s
2.7	Quantum Convolution (45 random parameters)	0.9045	0.9145	0.2637	0.2214	4600	707m 8s