Multi-class Classification using Quantum Transfer Learning

Bidisha Dhara*
Electrical Engineering
Indian Institute of Technology, Delhi
New Delhi, India
bidishadhara1@gmail.com

Monika Agrawal
Centre for Applied Research in Electronics
Indian Institute of Technology, Delhi
New Delhi, India
maggarwal@care.iitd.ac.in

Sumantra Dutta Roy
Electrical Engineering
Indian Institute of Technology, Delhi
New Delhi, India
sumantra@ee.iitd.ac.in

***Abstract*—Image Classification is one of the most important Machine Learning tasks, especially in this digital era. Though there exists classical algorithms which have performed quite well in multi-class classification tasks, classification using quantum architectures have mostly been limited to 2 or 3 classes. As the number of classes increased, the existing architectures did not achieve good accuracy. In this work, we aim to classify the MNIST dataset into 10 corresponding classes, using classical-to-quantum transfer learning. We performed both binary as well as multi-class classification using the hybrid architecture which yielded a maximum accuracy of approximately 100% and 90.4% respectively.**

I. Introduction

Quantum Systems are presenting an unprecedented era of computations with the use of subatomic level of particles to process data. Unlike classical computation here the information is being represented as qubits which is superposition of classical bits and hence they present much more processing power than their traditional counterparts. Quantum computers have shown a lot of potential even at its nascent stage, but now as the quantum hardware improves and this area is being explored to further extents to revolutionize the future of computational tasks.

With the size of datasets constantly growing and Moore's law coming to an end, we might soon reach a point where the current computational tools will no longer be sufficient. By carefully exploiting quantum effects such as interference or (potentially) entanglement, quantum computers can efficiently solve selected problems that are believed to be hard for classical machines. [1] Quantum machine learning extends the pool of hardware for machine learning by an entirely new type of computing device—the quantum computer. [2]

Presently hybrid architectures, or structures involving both quantum and classical computations, are also of much interest as Quantum computers are at very early stage. Hybrid set-up is typically a blend of "quantum" portion using a sequence of quantum gates operating on sets of qubits running on a quantum hardware, and a "classical" portion: with sequence of instructions running on a classical binary bits on regular computer.

Quantum Transfer Learning is one such hybrid architecture which can reap the benefit of both quantum and classical domain.

Transfer Learning is the technique of reusing a pre-trained model on a new yet related problem. Intuitively, it is much easier for a cyclist to learn to ride a bike, than someone who is untrained in cycling.

There have been few studies which have analysed Quantum Transfer Learning architectures. Mari et al. [3] have proposed different implementation of hybrid transfer learning, focusing mainly on the paradigm in which a pre-trained classical network is modified and augmented by a final variational quantum circuit. Their work has shown that "dressed" quantum circuit ("dressed" quantum circuit = classical pre-processing + quantum layer + classical post-processing) is a very flexible quantum machine learning model capable of classifying highly non-linear dataset. They have also explored the four different transfer learning approaches for classification tasks, namely:

1) Classical-to-Classical
2) Classical-to-Quantum for classifying Ant/Bees and Cats/Dogs
3) Quantum-to-Classical to encode 7 different 4×4 images by training only the final classical part
4) Quantum-to-Quantum for quantum state classification (gaussian/non-gaussian)

Classical-to-Quantum transfer learning is particularly appealing since it opens the possibility to classically pre-process large input samples (e.g., high resolution images) with any state-of-the-art deep neural network and to successively manipulate few but highly informative features with a variational quantum circuit. This scheme is quite convenient since it makes use of the power of quantum computers, combined with the successful and well-tested methods of classical machine learning [3].

Acar and Yilmaz [4] have used ResNet18 convolution network as feature extractor followed by a quantum variational classifier circuit to diagnose COVID-19 infected patients and normal (healthy) patients from Computerized Tomography (CT) images.

Soto-Pardes et al. [5] used the hybrid model of Quantum Transfer Learning to classify images of faces into classes identified as correct mask, incorrect mask and no mask. They reached an accuracy of 99.05% using ResNet18 as the classic

transfer learning model and a variational quantum layer of 4 qubits.
Gokhale et al. [6] implemented a hybrid model using ResNet50 pre-trained classical deep learning network and quantum variational circuit to classify spliced versus authentic images. Their comparative study showed an improvement in the quantum transfer learning approach (accuracy : 85%) from the classical implementation (accuracy : 80.57%)
Chalumuri et al. [7] proposed a quantum multi-class classifier as a variational circuit with a hybrid classical-quantum approach using quantum mechanical properties such as superposition and entanglement. They use this model to classify Iris dataset into 3 classes, Banknote Authentication dataset into 2 classes, Wireless Indoor Localization dataset into 3 classes achieving accuracy of 92.10%, 89.50% and 91.73% respectively.
It is observed, that though classical-to-quantum transfer learning approach is being explored, the work has mostly been restricted to binary classification tasks. We wish to extend the binary classification to a more generalized multi-class classification tasks. To the best of our knowledge this is the first attempt to use Quantum transfer learning for multi class classification. For this we studied the classification of the MNIST (Modified National Institute of Standards and Technology) dataset (handwritten digits) into 10 corresponding classes. We could achieve a maximum accuracy of 90.4% when classifying digits into 10 corresponding classes, while an accuracy close to 100% for several pairwise digit classification. Since using few epochs and limited data we could achieve this accuracy, we expect as the quantum resources improve, the accuracy will be higher. We use Pennylane [8] to simulate our models. Though we use simulators in this work, real quantum hardware has limitations on the number of qubits, thus we restrict ourselves to 4 qubits here.
Apart from this, we also wish to comprehend the behaviour of different quantum circuits for the same classification task.

II. Problem Formulation

Classifying handwritten digits with high accuracy has been a research interest for several decades, especially in this era of digital transformation. Classical Networks have performed quite well in this task with accuracy almost as the human brain. Islam et al. [11] implemented a multi-layer fully connected neural network with one hidden layer for handwritten digits recognition achieving an accuracy of 99.60% with test performance. Gope et al. [12] applied different classical machine learning algorithms such as Decision Tree Classifier, Support Vector Machine(SVM), Random forest classifier, Naive Bayes, K-Nearest neighbour algorithms to classify MNIST dataset with the highest accuracy of 95% achieved using SVM.
But not much has been explored in the quantum domain. We thus aim at comparing transfer learning models for binary classification, followed by designing models for multiclass classification using quantum computation.

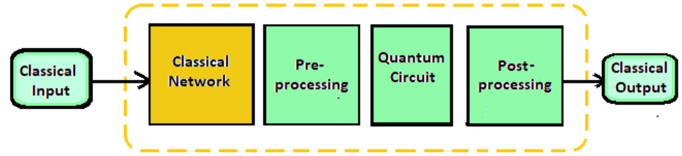


Fig. 1: A generic diagram of Classical-to-Quantum Transfer Learning Model .

A. *Transfer Learning*

Transfer Learning refers to the method of reusing a trained network for related but different task. This "transfers" the knowledge gained from one domain to another.
Here, we use the Classical-to-Quantum Transfer Learning approach to study the performance of the hybrid architecture for our classification task. A schematic diagram has been shown in Fig 1. The classical layer takes in input from the classical domain, produces some intermediate classical output, which is converted to the quantum domain. The quantum circuit then processes this output, which is again converted to classical measurement for human interpretation.
In this work, we have used ResNet18 as the classical Network, since it is one of the most popular CNN (Convolutional Neural Network) models for classification tasks. The ResNet18 network is a model pre-trained on ImageNet, from torchvision package. Although in this work, we use only ResNet18 as the classical network, it would be interesting to explore other versions such as ResNet50 in future work.

B. *Approach Used for Binary Classification*

We have designed hybrid models to classify each pair of digits for binary classification. For a total of 10 classes, we had 45 such pairs of digits. We also compare the results obtained by our binary classifier and the one shown in [3].

C. *Approach Used for Multi-class classification*

Since previous works ([4], [6]) have shown that the quantum transfer learning architecture have performed quite well for binary classification tasks, we use a heuristic to extend to multi-class classification, i.e. One_vs_Rest method. This method converts the multi-class classification task into multiple binary classification tasks.

III. MODEL

In this paper, we aim at designing a multi-class classifier using Classical to Quantum Transfer Learning approach. This sort of architecture has been previously used for binary classifications mostly. We propose a model to classify a set of images into two or more corresponding classes. For this work, we have classified the handwritten digits(MNIST dataset has been used) into 10 corresponding classes, though any dataset can be used.
Our research paper is grounded in Transfer Learning architectures, serving as the foundational framework for our work. Within this framework, we employ a hybrid classical-quantum

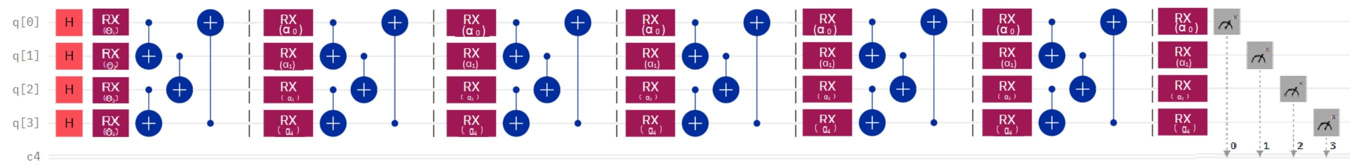


Fig. 2: The Quantum Circuit designed

circuit in conjunction with a one-vs-rest heuristic for multi-class classification. This combination of classical and quantum components forms the core of our data processing and classification methodology. The models we will further build are hybrid structures. So, a blend of classical and quantum portion exists as shown in Fig 1. We restrict the models to 4 qubits in this work. The quantum circuit takes in data from a classical predecessor, processes them, and then gives some output to the classical successor.

To elaborate, our approach involves processing input data in the form of images through a classical-quantum network. This network is specifically built upon a ResNet18 layer architecture, with the final layer being replaced by quantum circuits. This choice in architecture is a critical aspect of our theoretical framework, as it outlines how data is encoded into quantum circuits and how results are obtained through quantum measurements.

To test the efficacy of our approach, we have conducted experiments using binary hybrid models for each pair of digits, as well as an integrated multi-class classifier applied to a dataset encompassing all 10 digits. This rigorous testing methodology underscores the robustness of our framework.

We divide the work into two main sections, where in the first section we check the behaviour of this architecture towards classification of each pair of digits (i.e, binary classification) and in the second section we test different architectures on different simulators to achieve multi-class classification. Each of the models we further discuss are executed using simulators, with different hyperparameter settings. The images of MNIST dataset serve as the input to each of our models, while the network output is used to predict the output class using softmax activation function. For all the models discussed below, we mention the hyperparameters and simulators used, so that it is reproducible for future work.

A. Binary Classification

In order to understand the performance of this architecture with MNIST dataset, we first train our model with 2 classes at a time. We design a model using a quantum circuit and compare the output with the model used in [3]. The main motivation behind comparing the two models for the same classification task is to see how a change in the quantum circuit affects the classification results.

1) Quantum Binary Classification Model (QBCM): We build a model as shown in Fig 3 by replacing the "Quantum Circuit" block of Fig 3 with the circuit shown in Fig 2.

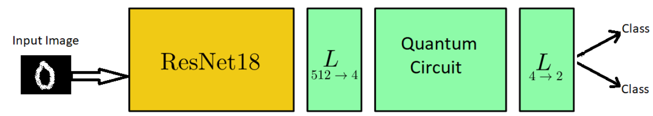


Fig. 3: Schematic diagram of model for binary classification for the pair of digits : 0 and 1. The binary classifier assigns probabilities to the 2 classes

In Fig 2, followed by the parallel Hadamard gates, the circuit has a set of Rotation gates which encode the input data. Primarily, this set of Rotation gates convert data from classical domain to quantum domain for further quantum computations to take place.

This encoding is followed by a variational circuit with adjustable parameters. The parameters in the circuit are the angles with which the Rotation gates are associated. While training this circuit, these parameters are updated at each iteration.

Finally we take a series of measurements of the qubits. These measurements are then used for further classical computations. The measurement of the qubits at the end of the quantum network is fed to a linear layer with mapping 4 $\rightarrow$2 which predicts the class.

The model shown in Fig 3 is for the two input classes 0 and 1. Similarly, models were built for each pair of input classes.

We train QBCM as well as the model used in [3] to classify 2 digits at a time. We take 2 classes, say 0 and 1, train the models with 1000 images of 0s and 1s, test them with 500 images of 0s and 1s. Similarly, we repeat this for every pair of digits for both the models.

The hyperparameters used for every model are:

- Optimizer : SGD
- Learning rate : 0.001
- Momentum : 0.9
- Number of epochs : 5

Both the models were simulated on Pennylane's default.qubit ([9]) ; a simple state simulator.

B. Multi-class Classification

Next, we move toward multi-class classification. Here, the transfer learning models built had used the quantum circuit of a binary classifier.

1) Quantum Multi-Class Classification Model-1 (QMCM-1) : Modifying only post-processing layer for multi-class classification: As a first attempt , the most obvious approach

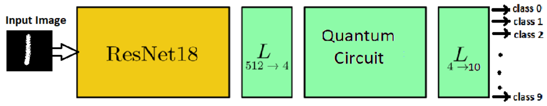


Fig. 4: Schematic diagram of QMCM-1 . The post-processing layer has been modified to assign probabilities to all 10 classes, instead of 2.

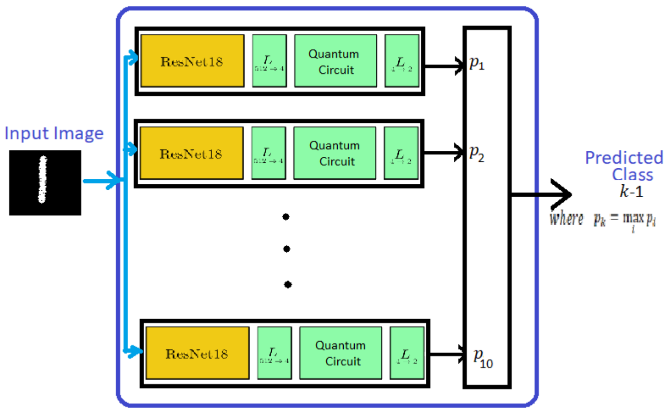


Fig. 5: Schematic diagram of the one-vs-rest model proposed

to extend the model shown in Fig 3 towards multi-class classification is to modify the post-processing layer of the network such that the model assigns probabilities to 10 classes instead of 2. We modify the post-processing layer to map $4 \rightarrow 10$ (since there are 4 qubits in the quantum circuit used) which is shown in Fig 4. The hyperparameters used were:

- Optimizer : SGD
- Learning rate : 0.001
- Momentum : 0.9
- Number of epochs : 5

We trained the model using 5000 images of all digits, and tested using 100 images. The model was simulated on Pennylane's default.qubit ([9]); a simple state simulator.

C. Quantum Multi-Class Classification Model-II (QMCM-2) : One_vs_Rest methodology

We present another approach here. We propose a one-vs-rest method which is a common technique to extend binary classifiers towards multi-class classifiers.
We first discuss the model architecture, followed by the experimental set-up used for simulations.

1) Model Architecture: We build the model in two parts:

a) Sub-models: We train 10 hybrid sub-models, such that each of them adapts to the classical-to-quantum transfer learning approach, where in the "classical" portion we use ResNet18 convolutional network with its fully connected layer replaced by the "quantum" portion: a quantum binary classifier. The ResNet18 network we used is a model pre-trained on ImageNet, from torchvision package.

Each of these sub-models are trained for individual digits, i.e, 1st model is trained to detect whether a given input is 0 or not 0, 2nd model is trained to detect a 1 or not 1 and so on. In other words, the ith submodel produces two outputs: p_{i0} and p_{i1} , where p_{i0} is the probability of a given input image to not be in class $(i-1)$ and p_{i1} is the probability that it belongs to class $(i-1)$; $i \in \{1, 2, ..., 10\}$

b) Integrating the sub-models: The output of the above 10 sub-models are further processed. Based on the 10 output of the 10 sub-models, each class of digit gets assigned a probability. The model finally predicts that class which has the highest probability assigned to it. The output from each of the sub-models are integrated using the Softmax activation function, which predicts the class of the input image.
For a given input image,
according to ith submodel, the probability that it belongs to class $(i-1)$ is given by p_{i1}.
We normalize these scores to use Softmax activation function. Thus, we assign probabilities to each class $(i-1)$ as $p_1, p_2, ...p_{10}$ where :
$p_i = \frac{p_{i1}}{\sum_{j=1}^{10} p_{j1}}$
Then, the predicted class = $k-1$, such that,
$p_k = \underset{i}{max} \quad p_i$ where $k \in \{1, 2, ...10\}$.
In other words, suppose the input image is that of the digit 5. Then, corresponding to this input, all the sub-models output some probability, i.e, 1st model outputs the probability by which the input image is a 0, the 2nd model outputs the probability by which the input image is a 1, and so on. Based on these outputs from 10 sub-models, the one who assigns highest probability is considered as the predicted class. In this case, it is expected that the 6th model will assign highest probability, since the input is that of digit 5. The schematic representation of the model is shown in Fig 5.

2) Experimental Setup: To train ith sub-model, we have used a subset of the MNIST dataset comprising of 300 images of i and 300 images of all digits. We tested each using 100 images of all digits.
The integrated model and the sub-models were simulated on Pennylane's default.qubit([9]); a simple state simulator, as well as qiskit.basicaer ([10]); a simplified version of the Aer device.
For the default.qubit simulator, SGD optimizer was used with hyper-parameters as :

- learning rate=0.001
- momentum=0.9
- epochs=6
- batch Size=10
- loss Function : Cross Entropy Loss

For the qiskit.basicaer simulator, SGD optimizer was used with hyper-parameters as :

- learning rate=0.001
- momentum=0.9
- epochs=6
- batch Size=10

TABLE I: Test accuracy achieved when training on 2 classes at a time using model shown in [3]. The entry in the ith row and jth column shows the accuracy on training with the classes i and j

	0	1	2	3	4	5	6	7	8	9
0	-	100	99.1	99.4	99.7	99.5	97.4	99.8	98.8	97.7
1		-	99.4	99.6	99.5	99.6	99.2	98.5	99.7	98.9
2			-	95.2	99.5	98.2	98.1	98.6	98.6	98.6
3				-	99.5	64.4	97.4	99	98.5	99.1
4					-	97.6	98.7	96.8	98.4	96.7
5						-	98.4	99.4	98.5	95.8
6							-	99.9	98	99.5
7								-	99.7	96.6
8									-	97.8
9										-

TABLE II: Test accuracy achieved when training on 2 classes at a time using QBCM. The entry in the ith row and jth column shows the accuracy on training with the classes i and j

	0	1	2	3	4	5	6	7	8	9
0	-	99.9	99.1	98.4	99.7	99.6	97.9	99.6	99.2	98
1		-	99.3	99.5	99.4	99.6	99.1	99.4	99.4	99.6
2			-	95.4	99.1	97.6	98	98.8	99	98.8
3				-	99.2	92.5	97	99.1	99.1	99.1
4					-	96.8	98.6	97.5	98.6	98.2
5						-	98.5	99.2	98.5	96.1
6							-	99.9	97.9	99.3
7								-	99.8	98
8									-	98.1
9										-

TABLE III: Accuracy achieved by the individual sub-models of QMCM-2 on $default.qubit$

Sub-model i									
1	2	3	4	5	6	7	8	9	10
98	99	90	98	98	98	98	93	94	95

TABLE IV: Accuracy achieved by the individual sub-models of QMCM-2 on $qiskit.basicaer$

Sub-model i									
1	2	3	4	5	6	7	8	9	10
98	100	92	98	99	96	99	95	96	97

TABLE V: Accuracy achieved by the proposed model QMCM-2 on $default.qubit$

	Target									
Prediction	0	1	2	3	4	5	6	7	8	9
0	83	0	4	0	0	1	15	2	5	1
1	0	126	0	0	0	0	0	1	0	0
2	0	0	109	6	1	1	1	4	0	0
3	0	0	1	98	0	6	0	7	11	2
4	1	0	0	0	105	0	1	1	1	0
5	1	0	0	2	1	78	2	0	2	0
6	0	0	0	0	0	0	67	0	0	0
7	0	0	2	0	1	1	0	84	2	4
8	0	0	0	0	0	0	1	0	68	2
9	0	0	0	1	2	0	0	0	0	85

TABLE VI: Accuracy achieved by the proposed model QMCM-2 on $qiskit.basicaer$

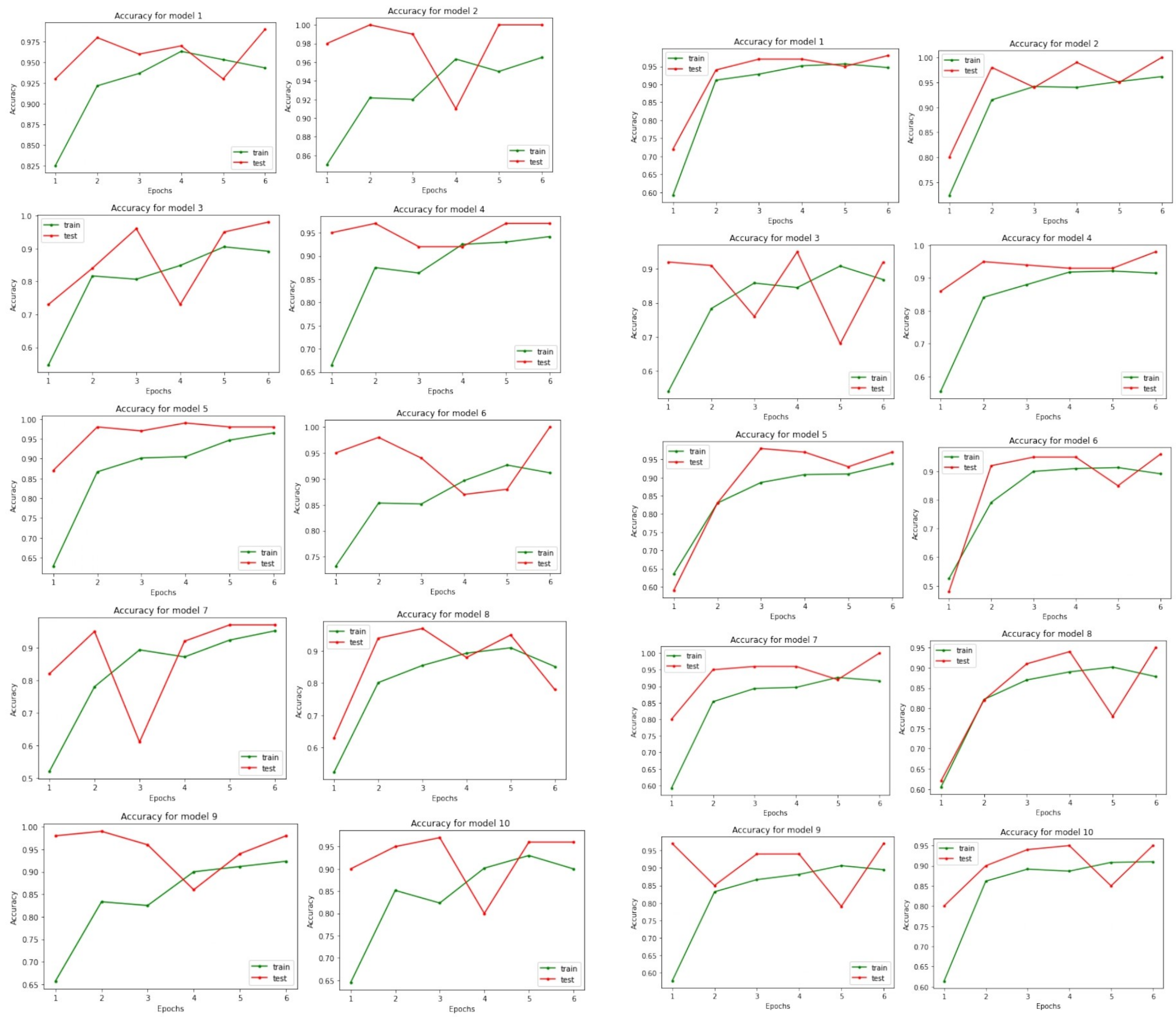
	Target									
Prediction	0	1	2	3	4	5	6	7	8	9
0	83	0	4	2	0	0	8	2	5	0
1	0	121	0	0	0	0	0	0	0	0
2	0	1	99	6	0	0	0	4	1	1
3	0	0	0	49	0	0	0	1	0	0
4	0	0	0	0	102	0	1	6	1	2
5	2	2	7	49	3	86	3	8	12	2
6	0	0	0	0	2	0	74	0	1	0
7	0	0	5	0	1	1	0	78	0	4
8	0	0	0	0	0	0	1	0	69	4
9	0	1	1	1	2	0	0	0	0	81

- loss Function : Cross Entropy Loss

Once all the sub-models were trained, the weights of the sub-models were frozen. These 10 sub-models were then integrated using a classical final layer.
The complete model was then tested using 1000 images consisting all 10 digits.
The learning curves for each submodel of QMCM-2 using $default.qubit$ simulator is shown in Fig 6a which depicts the performance of the submodels as they train over epochs for both train and test dataset. Fig 6b shows the performance of the submodels when they were made fully classical (using ResNet18 only). In simpler words, keeping the hyperparameters, training and testing dataset same, if we replace the hybrid architecture with ResNet18, we get the learning curves as shown in Fig 6b. But the plots of Fig 6b shows that 6 epochs is not enough to train the classical model. We train them for 2 more epochs. Fig 7 shows the curves when the fully classical submodels had been trained and tested for 8 epochs.

IV. Results and Discussion

In this section, we present the results obtained corresponding to each architecture and models discussed in the previous section. We present them in tabular form, for the ease of comparison between several architectures.
The binary classification results obtained corresponding to the model shown in [3] and QBCM are shown in TABLE I and TABLE II respectively. The results shown in these two tables clearly indicate that the network performs quite well for binary classification tasks. The $(i, j)th$ entry shows the accuracy in distinguishing classes i and j. Since distinguishing i, j and j, i are same, so the tables are strictly upper triangular. As also seen in [3], the architecture as QBCM performs satisfactorily when dealing with two classes.
Though the results of TABLE I and TABLE II are similar, a great improvement is achieved when training with classes 3 and 5. An accuracy of 64.4% was achieved using quantum circuit of [3], while it improved to 92.5% when the quantum circuit of Fig 2 was used.
QMCM-1 did not perform well enough. It has been shown in [5] that modifying the post processing layer to $4 \to 3$ gives



(a) (b)

Fig. 6: (a) shows the Training and Test accuracies obtained for each submodel corresponding to QMCM-2 using $default.qubit$ simulator. (b) shows the Training and Test accuracies obtained for fully classical models (using ResNet18) similar to each submodel of (a). For instance, the first plot of (a) shows the accuracies when the 1st hybrid submodel was trained to detect a 0 or not 0. Similar to that, the first plot of (b) shows the accuracies when a fully classical ResNet18 model was trained to detect a 0 or not 0.

good results, but in our case, changes made only to the post processing layer could not yield satisfactory results. The best accuracy obtained was 43%.

The results obtained corresponding to QMCM-2 are shown in TABLE III, IV, V and VI. TABLE III and IV show the accuracies obtained by the individual sub-models on default.qubit simulator and qiskit.basicaer simulator respectively. The entries are the accuracies obtained by the ith sub-model on $(i-1)$ class.

On integrating the sub-models together, TABLE V and VI show the accuracies of the complete model on default.qubit simulator and qiskit.basicaer simulator respectively. The entry in mth row and nth column shows the number of times n was predicted to be m. So, the diagonal entries are correct predictions, while off-diagonal entries are mis-predictions. Since the total number of test images were 1000, so the summation of all entries of the tables equal to 1000. Out of these 1000 test images, number of correct predictions is given by the summation of the diagonal elements of the table.

Comparing the accuracies obtained from both the simulators, it

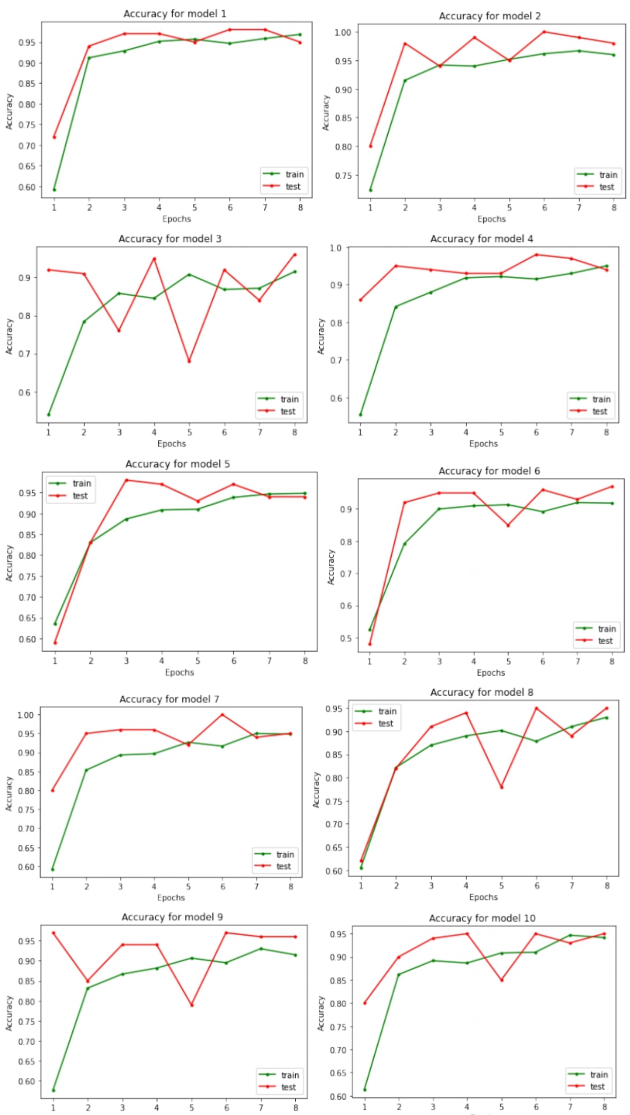


Fig. 7: Training and Testing curves for 8 epochs (instead of 6) for same submodels as shown in Fig 6b

is clear that though the diagonals for both cases are dominant, (the correct predictions represent the diagonal), default.qubit achieves an accuracy of 90.4% while qiskit.basicaer achieves 84.2% accuracy in this classification task. It is also to be noted, that we could achieve this accuracy with as little as 6 epochs.
In this architecture, though the number of models to be trained are high, but the results are quite satisfactory as compared to the classical architectures. Except a few off-diagonal mispredictions, most of the predictions are correct. Also, the quantum architecture requires lesser number of epochs to train than the classical counterpart. Thus, we achieve better accuracy with less data and less training time.

V. Conflict of Interest

The authors have no conflicts of interest to declare. All co-authors have seen and agree with the contents of the manuscript and there is no financial interest to report.

VI. Data availability statement

The MNIST (Modified National Institute of Standards and Technology) handwritten digit database can be found in the link below: http://yann.lecun.com/exdb/mnist/

References

[1] C. Ciliberto, M. Herbster, A. D. Ialango, M. Pontil, A. Rochetto, S. Severini, L. Wossnig (2018) “Quantum machine learning: a classical perspective”
[2] https://pennylane.ai/qml/whatisqml/
[3] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran,(2020) “Transfer Learning on hybrid classical-quantum neural networks ”
[4] E. Acar, I. Yilmaz (2020) “COVID-19 detection on IBM quantum computer with classical-quantum transfer learning”
[5] C. Soto-Paredes, J. Sulla-Torres, (2021) “Hybrid Model of Quantum Transfer Learning to Classify Face Images with a COVID-19 Mask”
[6] A. Gokhale, M. B. Pande and D. Pramod (2020) “Implementation of a quantum transfer learning approach to image splicing detection”
[7] A. Chalumuri, R. Kune, B. S. Manoj (2021) “A hybrid classical-quantum approach for multi-class classification”
[8] https://pennylane.ai/
[9] https://docs.pennylane.ai/en/stable/code/api/pennylane.device.html
[10] https://docs.pennylane.ai/projects/qiskit/en/latest/
[11] K. H. Islam, G. Mujtaba, Dr. R. G. Raj, H. F. Nweke (2017) “Handwritten Digits Recognition with Artificial Neural Network”
[12] B. Gope, S. Pande, N. Karale, S. Dharmale, P. Umekar (2020) “Handwritten Digits Identification Using Mnist Database Via Machine Learning Models”