Histopathological Cancer Detection

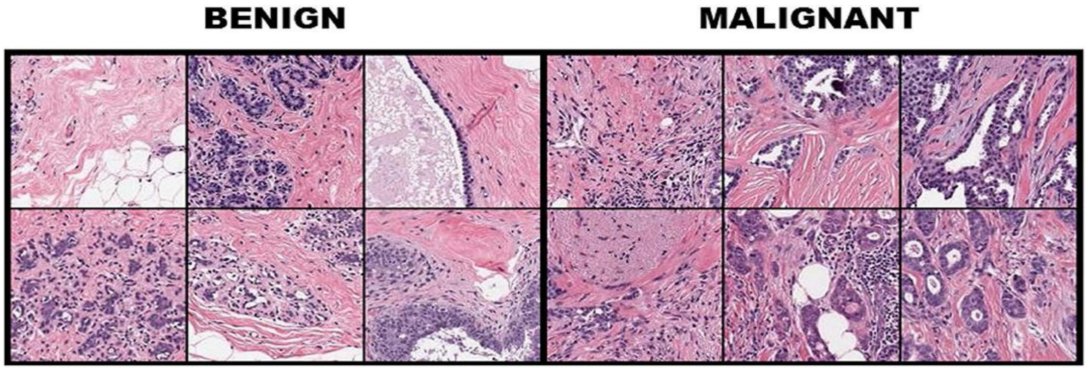
Team ERS_CRRCT23

Meet the Team!

- Ramachandran SS
- Elizabeth Grace
- Satzhan
- Sajed

Introduction

- The analysis of histopathological (study of abnormal or diseased cells) phenotypes plays a key role in cancer research and medicine; it remains a challenge to define these phenotypes.
- These analyses are vital to link molecular mechanisms with disease prognosis.
- Owing to the complexity of cancer, its different morphologies, spatial arrangements and behaviors : multiscale histopathology features need to be extracted using deep learning methods.
- A hybrid Quantum Computing approach increased the accuracy of histopathological cancer classification to 95%.



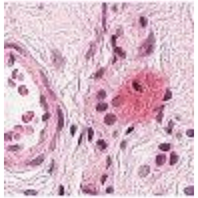
Approaches evaluated

- Classical Deep neural networks.
- Quantum Hybrid machine learning using classical feature extractor.
- Quantum Hybrid machine learning using convolutional autoencoders.
- Quantum Convolution based encoder and classical classifier.

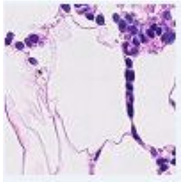
Dataset preparation

Dataset was taken from Kaggle :
https://www.kaggle.com/competitions/histopathologic-cancer-detection/data

A cancerous image



A non cancerous image



The whole dataset contains 220,025 images. It is divided into train set (75%) and validation set (25%).

Train Set: 165,018, Validation Set: 55,007.

Classical Deep learning

The images are passed to convolution neural networks which extracts the features and final those features are being sent to a fully connected layer with softmax function which performs the classification.

The model is trained using the PyTorch framework with CUDA for parallelization.

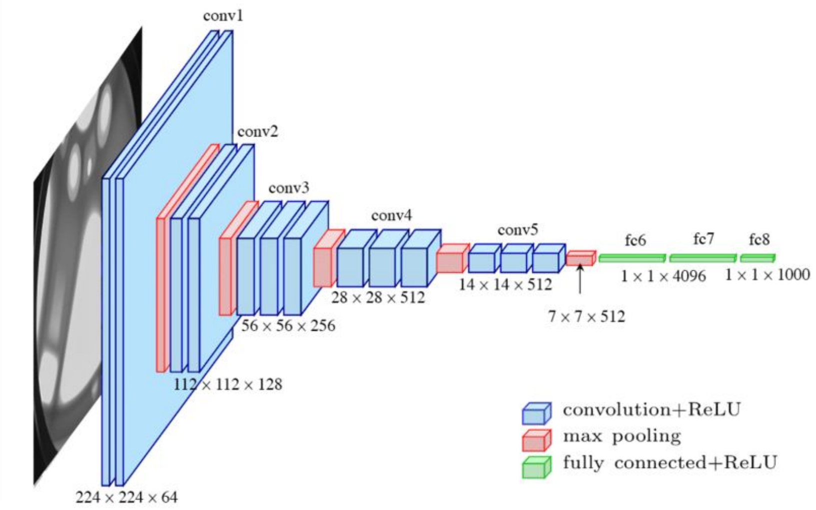
Models

VGG-16
ResNet-18

Hyperparameters

Batch size: 16
Learning Rate: 1e^-5
Optimizer: Adam
Epochs: 15

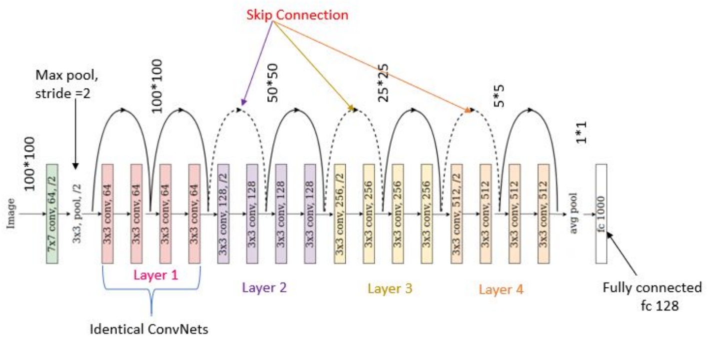
Model Architecture - VGG16



The model uses a set of convolution layers and max pooling layers to extract the features.

The final set of features are flattened and being passed to the Linear classifier.

Model Architecture: ResNet-18



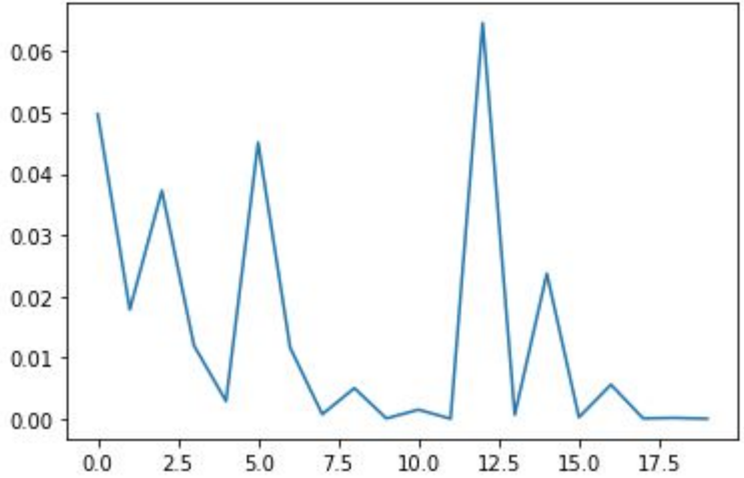
ResNet-18 Architecture

Fruit 360 Input Image size= 100*100 px

The ResNet-18 model is a very deep neural network model compared to VGG-16 with a huge set of parameters.

In order to avoid the problem of vanishing and exploding gradients the model is equipped with skip connection layers which moves the gradient by skipping layers while backpropagation to avoid these problems.

Results - Classical Deep Neural Networks- VGG-16



L
o
s
s

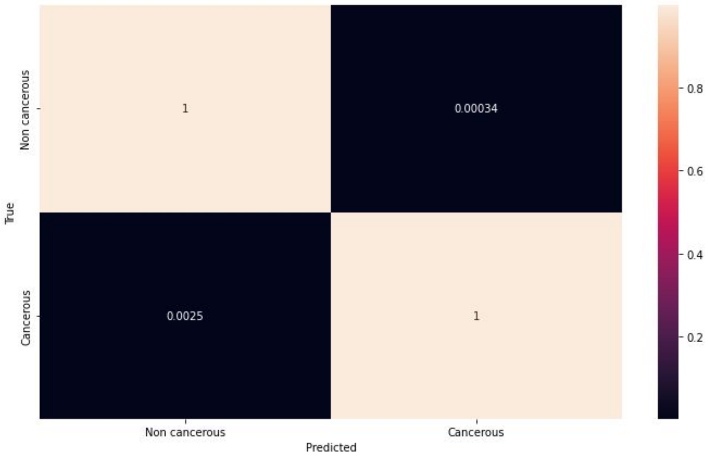
Epochs

The VGG-16 model trained with hyperparameters mentioned before was able to achieve around 96% accuracy in the test set.

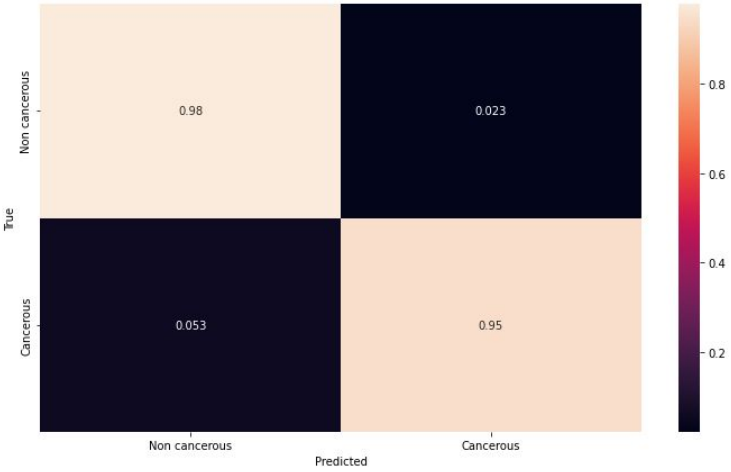
Epoch = 15/25, Train Acc: 99.87, Val Acc: 96.47

Confusion matrices

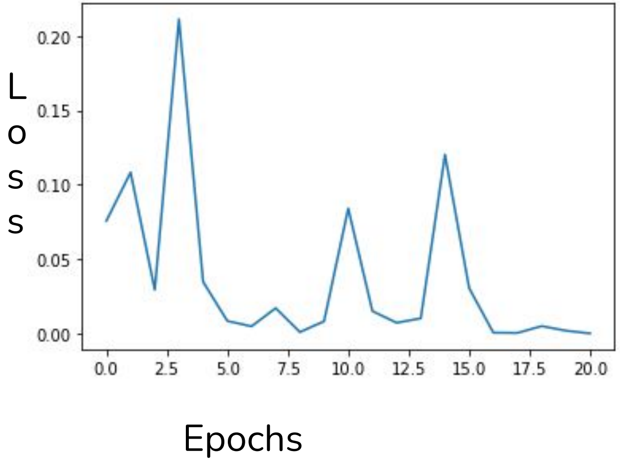
Train CM



Test CM



Results: Classical Deep Neural Networks ResNet-18

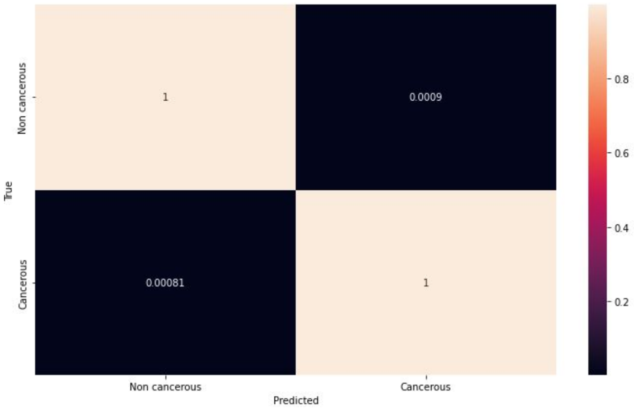


The ResNet-18 model was trained with the same hyperparameters mentioned before and it was able to achieve around 94% accuracy on the test set.

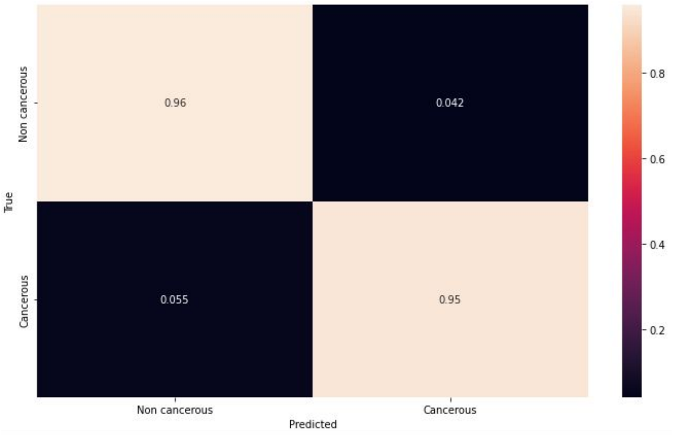
Epoch = 15/25, Train Acc: 99.50, Val Acc: 94.51

Confusion matrices

Train CM



Test CM



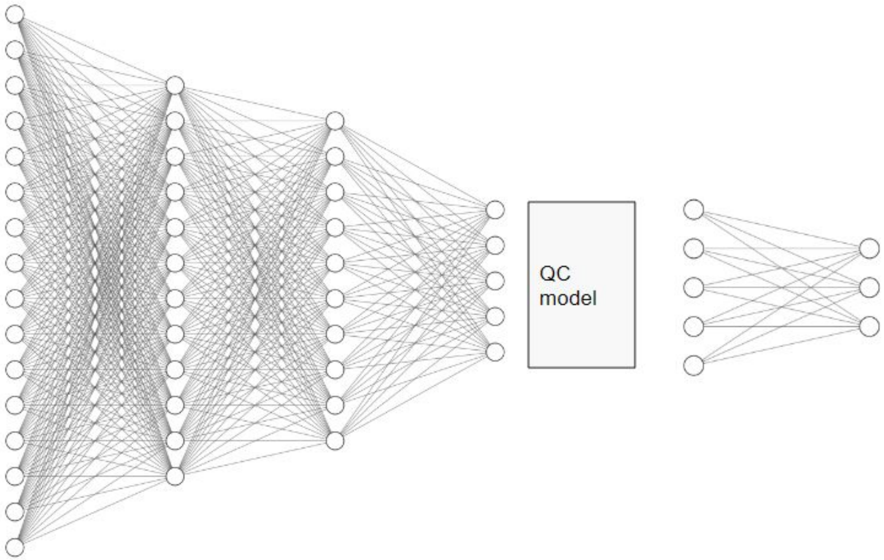
Feature Extractor

The ResNet-18 model for classification problems is trained to extract features of the input image using which the linear layers performs the classification.

The right hand side images shows the images which are the inputs to the model and the left hand side images are the features which are identified by the model to perform the classification.

Quantum Hybrid Model using feature extractor.

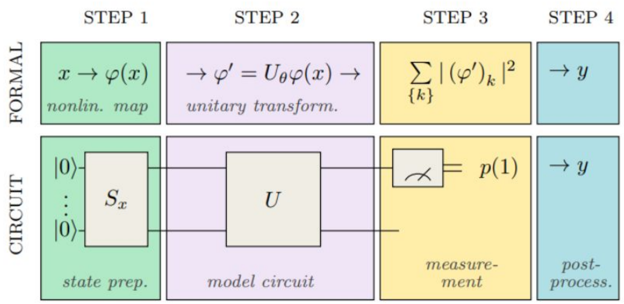
The ResNet-18 model is used to extract features from the image and the quantum model acts as a classifier.



Quantum model

The quantum model contains 4 parts

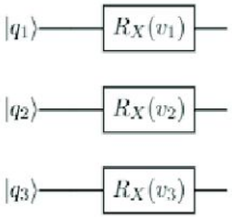
- The State preparation circuit which prepares the initial state to be passed.
- The Quantum model which contains the below components.
  - The Embedding circuit, which converts from classical to Hilbert space.
  - The Ansatz, which is the model with trainable parameters.
- The Measurement circuit which converts back from Hilbert space to classical
- The post processing circuit.



Quantum Model

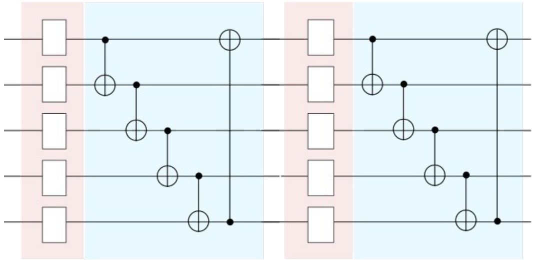
Embedding circuit

The input features are normalized to -1 and +1 using tanh activation function and passed as angles to the rotation gates to convert into Hilbert space.

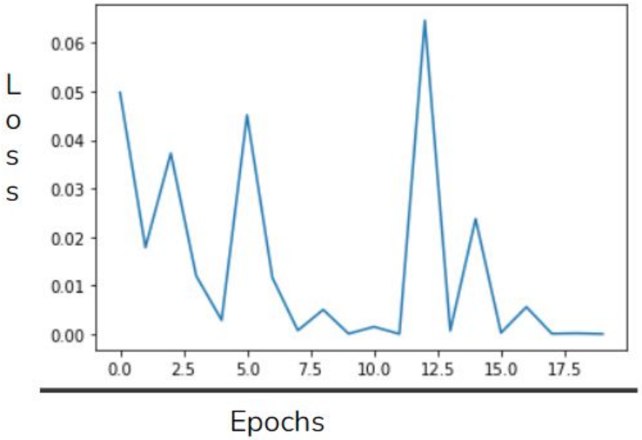


Ansatz:

The model uses a basic entanglement circuit with trainable parameters for the rotation gates as the ansatz.



Results - Pennylane Simulator

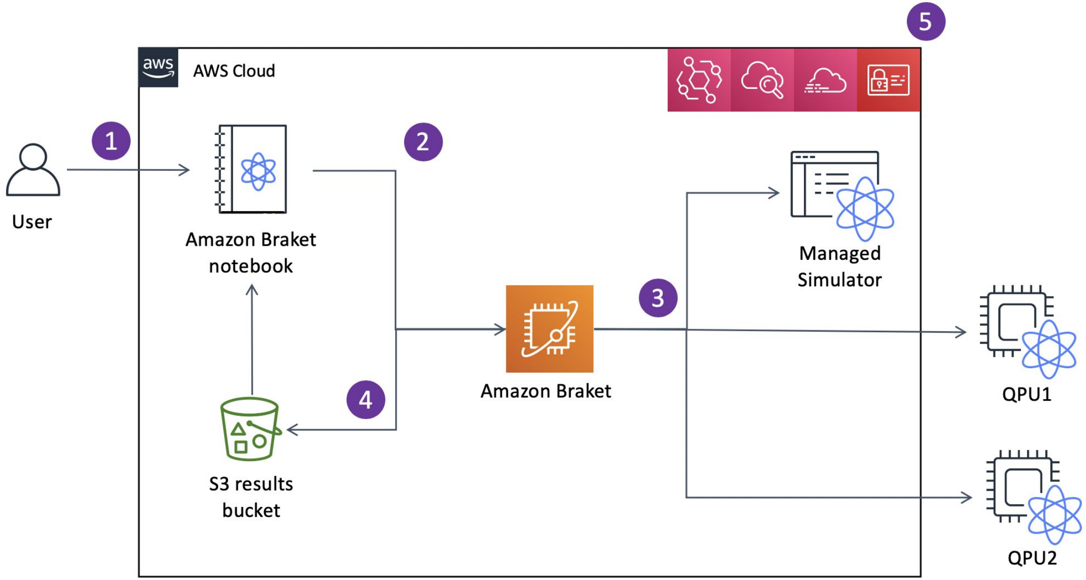


The feature extractor (ResNet-18) is also trained along with the model using the same hyperparameters as before and it was able to achieve around 95% accuracy in 14 epochs.

Epoch: 14/25 Train Loss: 0.2659 Train Acc: 99.1474 Val Acc: 94.8625

Results - AWS Braket - Rigetti — Aspen-M-3

The model was also run on a Rigetti Quantum computer provided by AWS which has 79 qubits. The AWS architecture is mentioned below



The train and validation datasets are stored as objects in S3.

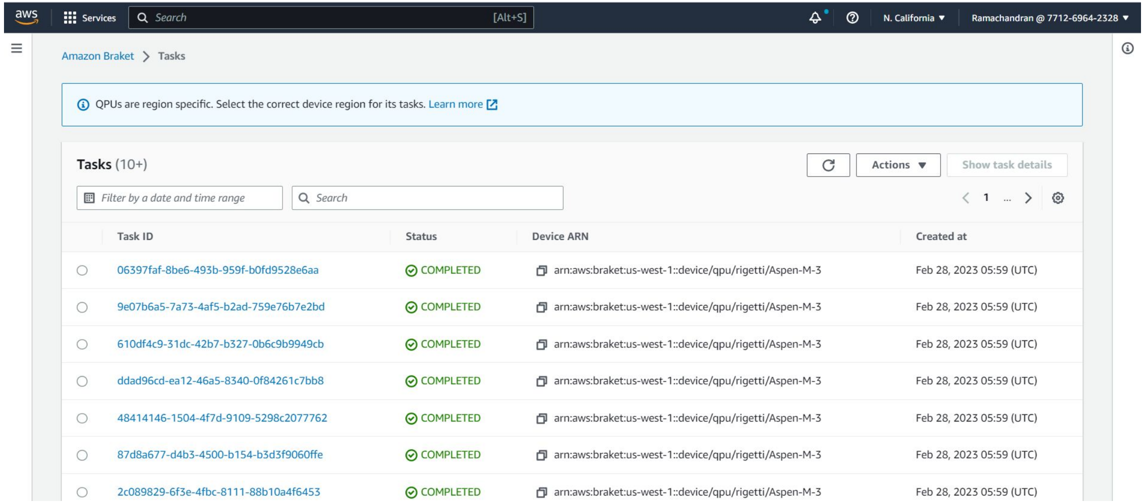
The pytorch trainloader performs the conversion from S3 objects to tensors.

The Classical feature extractor is run on AWS EC2 instance.

The features along with the quantum circuit are sent as Braket jobs to be run on Rigetti QPU.

The Resulting model parameters are stored in S3 bucket.

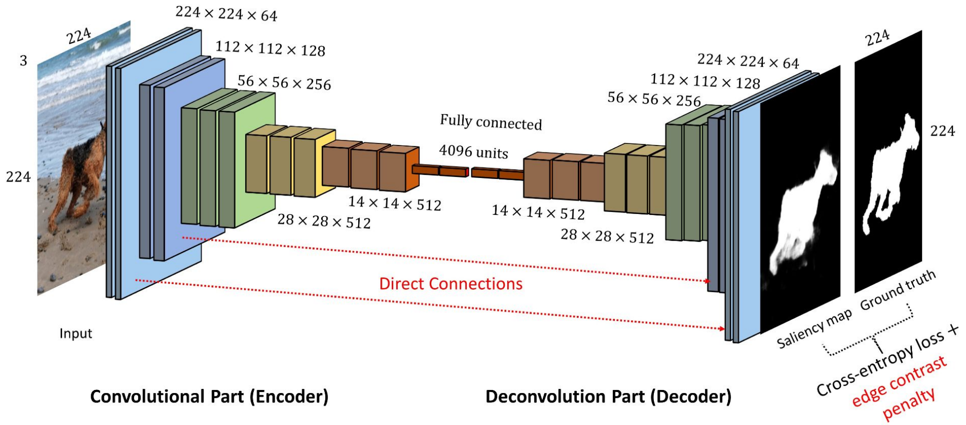
Results - AWS Braket - Rigetti (contd).



Since each gradient computation requires parameter shiting and rerunning the quantum circuit, a single iteration took a long time. Thus, we were only able to run it for a single iteration and for the rest of epochs the model was trained in simulator.

Autoencoders

The autoencoders take the input image and provided a reduced representation of the image which can later be used to obtain the input image by using the same deconvolutions as the encoder.



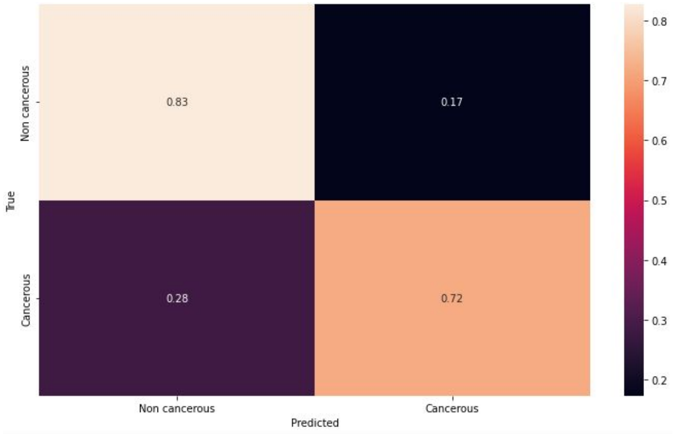
Quantum Hybrid Machine learning using autoencoders - Results

The output of the autoencoder is flattened and provided to the quantum hybrid model. The quantum hybrid model contains a set of linear neurons which reduces the feature size to number of qubits and then the same quantum model as previous is used to perform the classification.

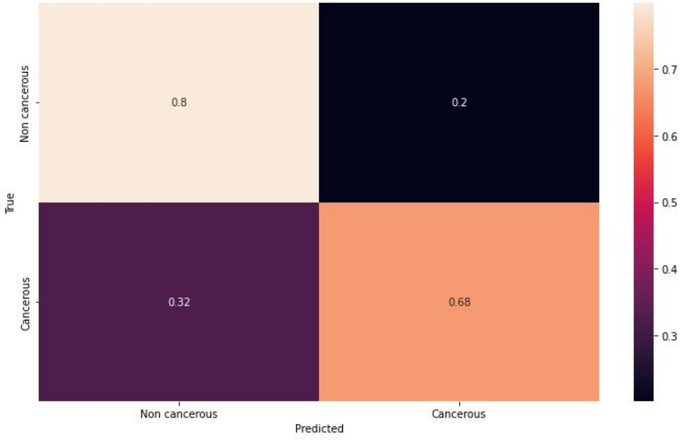
Epoch: 15/25 Train Loss: 0.2710 Train Acc: 78.2375 Val Acc: 74.9450

Confusion Matrices

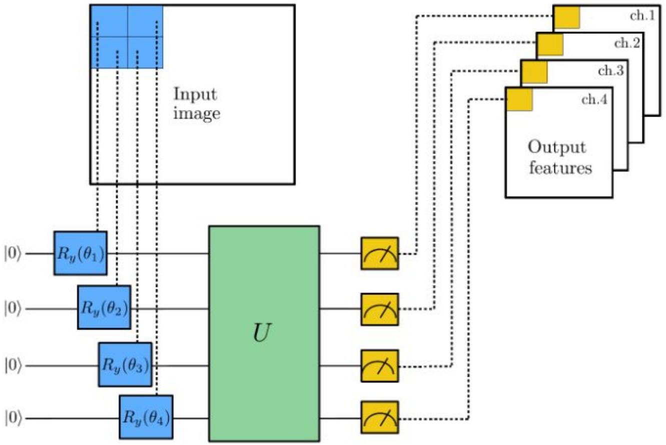
Train CM



Test CM



Quanvolutional neural nets

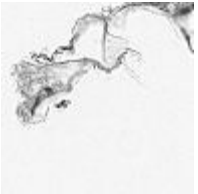


The 2x2 filter is being applied to the input image and moved with a stride of 2.

The pixels in those filters are being passed to quantum model and the 4 output are stored in separate images.

Results - QNN (Using basic entangler)

Input image -

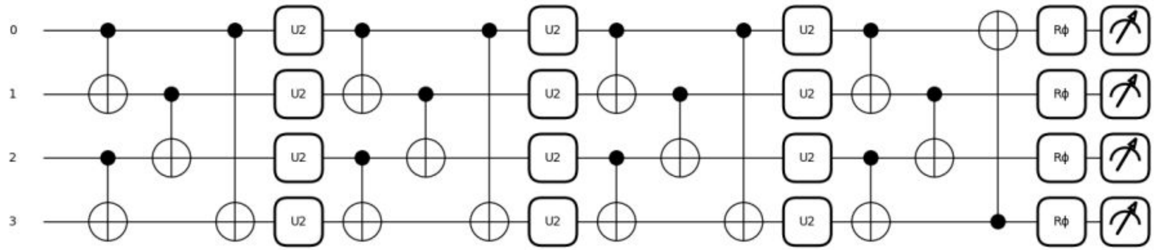


The 4 output features -

The same angles which are used for angle encoding are being used in the basic entangler.

But the output features are scattered very much which makes it difficult to train.

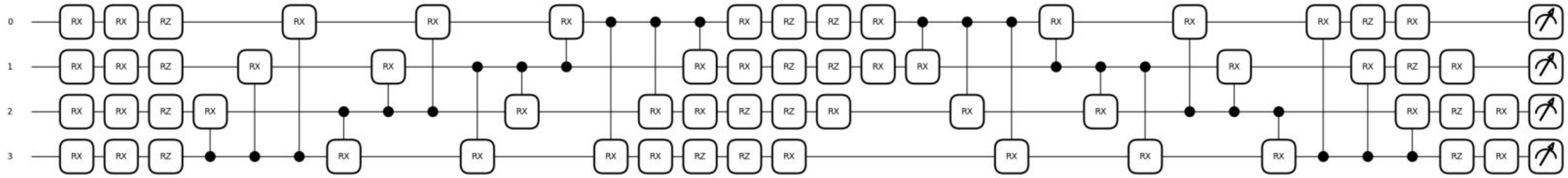
Other Ansatz explored



The unitary gates, each of them takes 3 parameters which makes it much difficult while computing gradients using AWS braket Rigetti, as each of the parameter shift must be evaluated for each unitary gate.

Since there are 12 unitary gates, we need to shift each gate's parameter 3 times in both directions leading to a circuit of size 12*3*2*(# shots) to be evaluated.

Other Ansatz explored - contd.



This circuit contains 60 parameters, so it takes a lot of time for AWS Braket Rigetti to perform the training.

Summary

Classical ResNet-18 model has a accuracy of 94% compared to VGG-16 which is of 96% on the test set. The reason for this is that huge models like ResNet-18 are prone to overfitting.

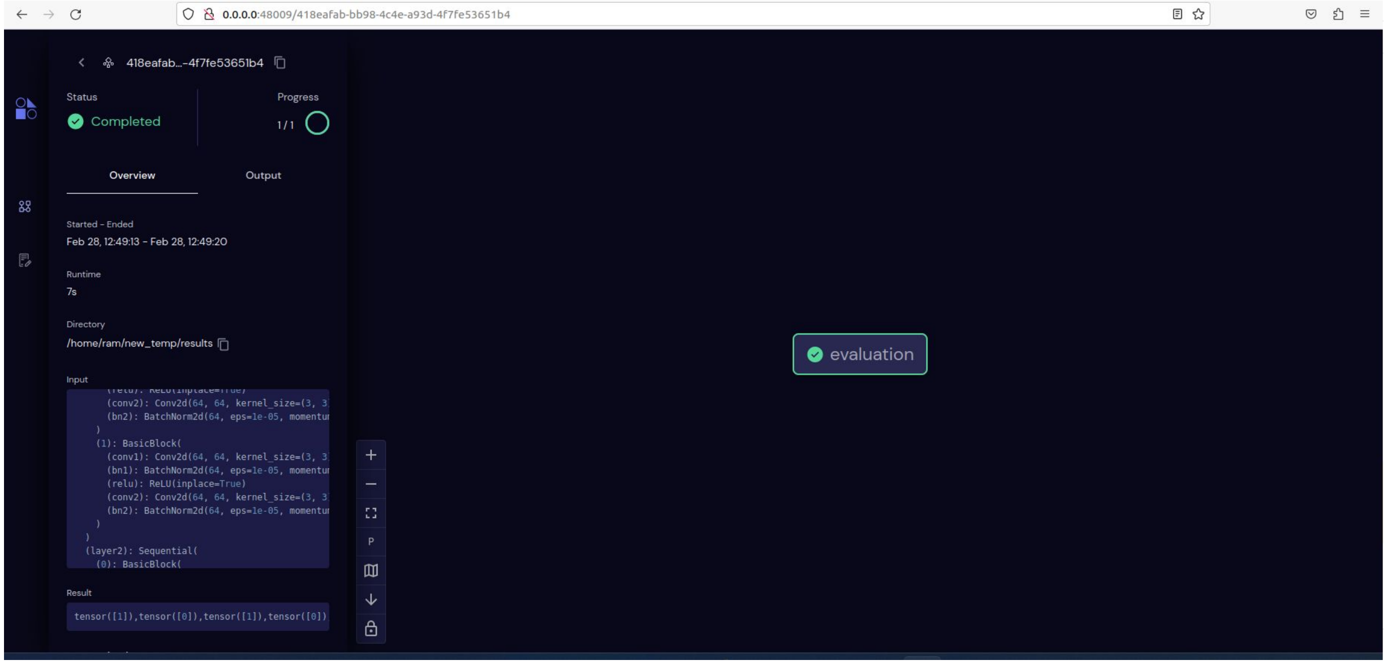
Using a Quantum hybrid model with ResNet-18 as feature extractor, we were able to achieve 95% accuracy, close to VGG-16 on the test set. This hybrid model reduces the overfitting of the classical model.

Using a Quantum hybrid model with convolutional autoencoder is not helpful for this dataset as it reduces the test accuracy to 78%.

Using Quantum Neural nets which perform Quanvolutions provides a quantum encoding representation, but after that is converted to the classical format for training, it appears to be scattered thereby affecting training.

Using the huge ansatz, as mentioned before, increases the training time without much improvement in accuracy compared to our Basic Entangler. So the Quantum Hybrid model with classical feature extractor and basic entangler perform better compared to classical ResNet-18 and can run faster in a real machine compared to the other ansatz which uses Unitary gates or a large number of rotation gates.

Covalent - Quantum hybrid model deployment.



The data is stored in Images folder and the covalent flow generates the output class 1- cancerous, 0 - non cancerous.

References

https://arxiv.org/pdf/2302.04633.pdf
https://www.kaggle.com/competitions/histopathologic-cancer-detection/data
https://docs.aws.amazon.com/braket/latest/developerguide/braket-what-is-hybrid-job.html
https://covalent.readthedocs.io/en/latest/tutorials/1_QuantumMachineLearning/pennylane_hybrid/source.html
https://pennylane.ai/qml/demos_qml.html

Thank You!