Project #12: Quantum Autoencoder and QML used on Medical data

Mentors:

- Alberto Maldonado-Romo
- Daniel Sierra-Sosa

Mentees:

- Pranshi Saxena
- Tan Jun Lian

Motivation

There is a steady increment in the incidence of Colon cancer in recent decades, this has led to a growing number of medical tests, being colonoscopy the standard test. This has generated an increase in medical personnel workload, which sometimes leads to specialists finding it hard to keep up the health care system.

There isn’t much related study been done, there is still a lot of potential and possibility for different method how we can apply quantum machine learning in medical anomaly detection and in this case to detect polyps during colonoscopy.

It's bit difficult to identify polyps with human eye especially in the early stages of its growth, the motivation is to create a QML Technique that can be used to detect and stop the growth of colon polyps in its zeroth stage.

Objective

Explore different method how we can implement quantum autoencoder and compare them with other anomaly detection techniques in doing segmentation task.

Perform benchmarking of different Dimension Reduction/Feature Selection algorithms in both Classical and Quantum Hardware, and evaluate the the potential of Quantum Machine Learning in NISQ in comparison to Classical Techniques.

Develop a QML Algorithm to detect colon polyp in its early stage of growth to further prevent its spread which can increase the risk of cancer.

Data Source

There are three polyps database, a proprietary and two public(CVC-ClinicDB & ETIS-LaribPolypDB) extracted from colonoscopy videos. The frame contain example of polyps with the ground truth which consists region covered by the polyps. CVC-ClinicDB has been generated from 25 different video studies. For each study at least a sequence containing a polyp was extracted. Considering this, CVC-ClinicDB database is composed by frames from 29 different sequences containing a polyp, is the database to be used in the training stages of ISBI 2015 Challenge on Automatic Polyp Detection Challenge in Colonoscopy Videos.

CVC Colon Database:

- 380 colonoscopy images containing polyps.
- The image size is 500x574 pixels.
- The images have been taken from 15 colonoscopy videos
- the database provides the detection masks and polyp contours, enabling the evaluation of the
- segmentations performed.
- This database has been created by expert endoscopists
- This database is open access

Project Plan

Qiskit

- Step 1: Research and understand current research field in anomaly autoencoder and dimension reduction/feature selection and segmentation techniques for Healthcare data, or other possible techniques/methods
- Step 2: Implement them
- Step 3: write a manuscript with our result

1. Method :
   - Autoencoder
   - PCA
   - other data compress method like Huffman coding, Golomb coding, etc.[1]

2. Proposed Idea:
   - Discrete (latent) representation with disentangled segmentation, for example: VQVAE classify itself few times and ecode the most similar vector from a list(c.f self-attention), so it won't have unlimited possibility of encode vector.
   - Other pre-trained model, model and techniques...
   - Applying CNN and QNN Models for Segmentation

Analysis:
- Swaptest
- Different Quantum Image Processing tecchniques[2]
- Model performance, and other benchmarking like image lossy analysis, etc.

reference:
[1] https://www.nature.com/articles/s41598-021-91920-x
[2] https://arxiv.org/pdf/2203.01831.pdf

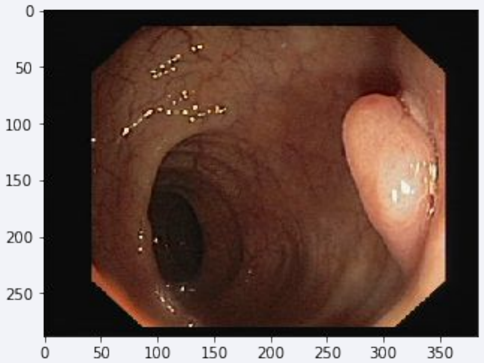
Dimension Reduction Techniques

The first step of our project was to explore and do research to understand the current research field in anomaly autoencoder and dimension reduction/feature selection and segmentation techniques for Healthcare data, or other possible techniques/methods.

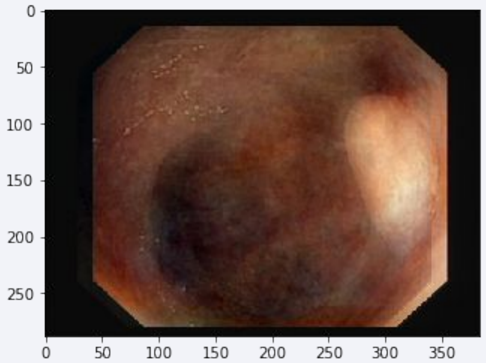
We have studied different research techniques and algorithms in both quantum and classical machine learning, and now we are working on implementing it on our medical data set. We have currently implemented different techniques for dimension reductions using classical machine learning algorithms.

Principal Component Analysis

- Here are the results of applying PCA to our CVC-ClinicDB database, Original Images. Original Data Set with size: 331776, shape: (288,384,3)



Original Data Set with
size: 331776, shape: (288,384,3)



Reconstructed Image from PCA
Data Size Dimension reduced to 100

The Explained Variance Ratio : 0.933

Reconstruction Error (MSE) came : 0.0015

Segmentation Techniques

Convolutional Neural Network (Defined Architecture)

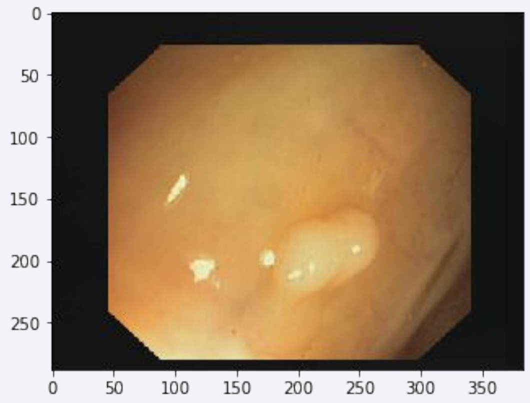
It is a U-Net-like architecture commonly used for pixel-level semantic segmentation tasks. Here is the detailed description of the Model

The model consists of several convolutional layers (conv1 to conv6). Each convolutional layer applies a 3x3 kernel to the input feature map, followed by a Rectified Linear Unit (ReLU) activation function.

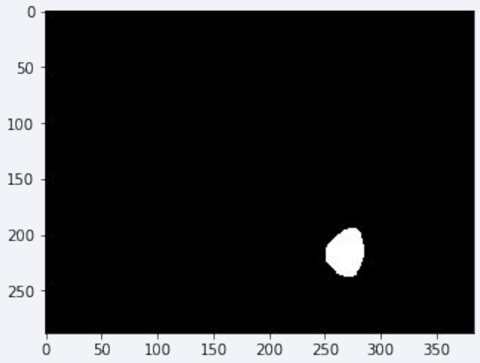
Max pooling (maxpool) is performed after every two convolutional layers, with a kernel size of 2x2 and a stride of 2. This downsamples the feature maps, reducing their spatial dimensions while preserving important features.

The final convolutional layer (conv6) outputs a feature map with num_classes channels. It uses a 3x3 kernel and does not apply any activation function after it.

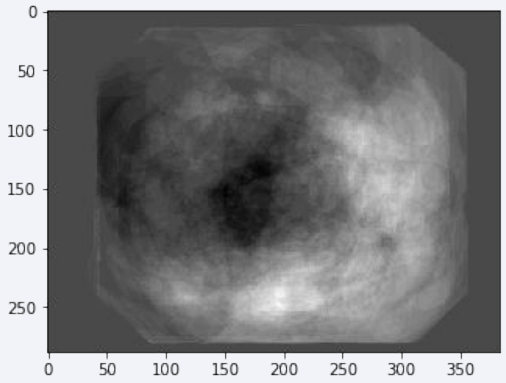
The upsample layer performs bilinear upsampling with a scale factor of 16. It increases the spatial dimensions of the feature map to match the input image size. The forward method defines the forward pass of the model. It applies the convolutional and pooling operations sequentially and returns the upsampled output.



Original Data Set with
size: 331776, shape: (288,384,3)



Ground Truth Data Set with
size: 331776, shape: (288,384,3)



Output Segmented Image(Gray Scaled)

Segmentation Techniques

Qiskit

Quantum Neural Network (QNN) (Defined Architecture)

This model combines classical neural network layers with a quantum circuit to perform computations.

In this model, the number of input neurons equal to the number of qubits and the number of output neurons equal to the number of classes.
It takes an input tensor x and performs operations on a quantum circuit (qc).

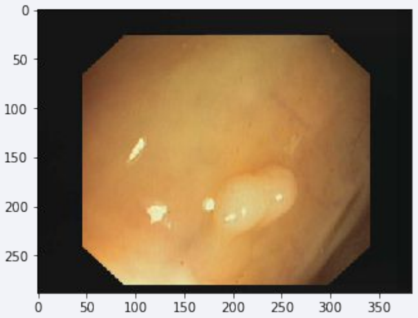
For each input sample, the method applies a rotation gate (ry) with angles calculated from the input tensor. The angles are scaled by π before applying the gate. This operation allows the input data to be encoded into quantum states.

After encoding the input, the circuit performs measurements (measure_all()) to obtain the quantum states' classical outcomes.

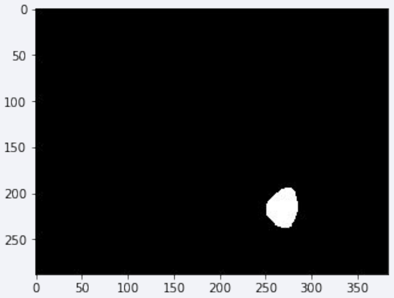
The circuit is then executed on a quantum simulator (qasm_simulator) using Aer from Qiskit. The transpile function translates the quantum circuit into a format that the backend can execute, and the assemble function prepares the job for execution. The result of the simulation is obtained using backend.run(job).result().

The measurement outcomes are extracted from the result, and the counts of different measurement values are recorded.

The probabilities of different measurement outcomes are calculated based on the counts obtained. These probabilities represent the quantum states' likelihoods after measurement.The probabilities are returned as a torch tensor of type torch.float32.



Original Data Set with
size: 331776, shape: (288,384,3)



Ground Truth Data Set with
size: 331776, shape: (288,384,3)

Output Segmented Image
(Coming Blank Screen)