UNIVERSITY OF PORTSMOUTH

School of Computing

M34099-QUANTUM COMPUTING

Pneumonia Detection on X-Ray scans using ResNet and VQC

Date of Submission:

07/12/2025

Module Lecturers:

Dr Fahad Ahmad

Student ID:

UP957417

Pneumonia Detection on X-Ray using ResNet and Quantum Computing

1. Introduction

In general quantum and classic computing were developing differently, however with current trends and plateau in developing stronger and more powerful processors, quantum computing started to gain more attention. Modern computing essentially began with the introduction of integrated circuits(IC) in 1958-1959 by Jack Kilby and Robert Noyce [1]. [2] summarised that developers tried to fit electronics with complex functions into a limited space and make it as light as possible. By 1965 integrated circuits were already established as mandatory for some government structures, such as military or space programs, due to reduced production costs and effectiveness, combined with less space and weight requirements. The main argument in [2] is that Gordon Moore observed the amount of components(resistors, transistors etc.) had been doubling each year and predicted this trend to continue for at least next 10 years:

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year
(see graph). Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least ten years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65 000." (Moore, 1998)

The graph [2] was referring to is on Figure 1.

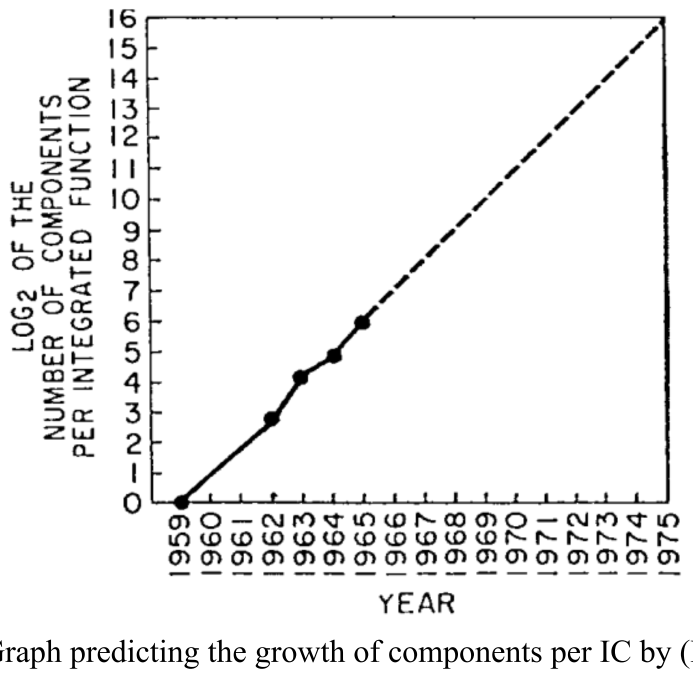


Figure 1. Graph predicting the growth of components per IC by (Moore, 1998).

This phenomenon is now known as Moore’s Law. And the clock speed of processors kept going higher and higher up until the 2000s, when the clock speed had stuck at around 3,5 GHz.
In the meantime, quantum physicists started applying quantum physics principles to computing algorithms, swapping classic bits with quantum bits. The earliest example of quantum computing is the quantum Turing machine presented by Paul Benioff in 1980 [3][4]. And around this time the quantum computing field gained its initial momentum along with computational power gain, when physicists started to simulate quantum dynamics. Yuri Manin and Richard Feynman suggested it is better to use hardware built using quantum phenomena [3][4].
Quantum algorithms proved their efficiency. Such algorithms as Shor’s algorithm for large-number factorisation and Grover’s algorithm in unstructured search problems demonstrated quantum computing potential in improving time complexity and optimising performance of various classic computing algorithms. Therefore, quantum computing affects multiple fields and machine learning is one of them, thus the **quantum machine learning**(QML) field has occurred.
[3] explained general ideas behind quantum kernels, quantum convolutional neural networks (QCNNs) and fault tolerant quantum transformers were developed.
There are also such methods as hybrid models, that use both classic and quantum machine learning principles as a solution. In this paper Residual Network(ResNet) by [5] is combined using a custom quantum layer at the end before a fully-connected layer used for classification as a main solution. The dataset used for fine-tuning could be found on Kaggle [6] - a pneumonia detection dataset among pediatric patients aged 1-5 on X-Ray images. This paper will let us understand the effects of applying quantum computing to well-established classic solutions in hybrid form.

2. Literature Review

2.1 Classic Computer Vision solutions.

2.1.1 Classic machine learning

Support Vector Machines(SVMs) thoroughly described by [4] are one of the most researched machine learning algorithms. The main idea is that in classification linear data may be separable by a hyperplane (Figure 2). And such principles are also used in regression tasks. For non-linear data, a kernel trick(mapping data to a higher dimension, Figure 3) was invented [16][17].
Support Vector Machines also perform very well on text and image classification tasks. [8] focuses its research on applying different learning techniques using SVMs for image classification tasks. But algorithms and models based on Convolutional Neural Networks showed to be better and more optimised for the task of image classification.

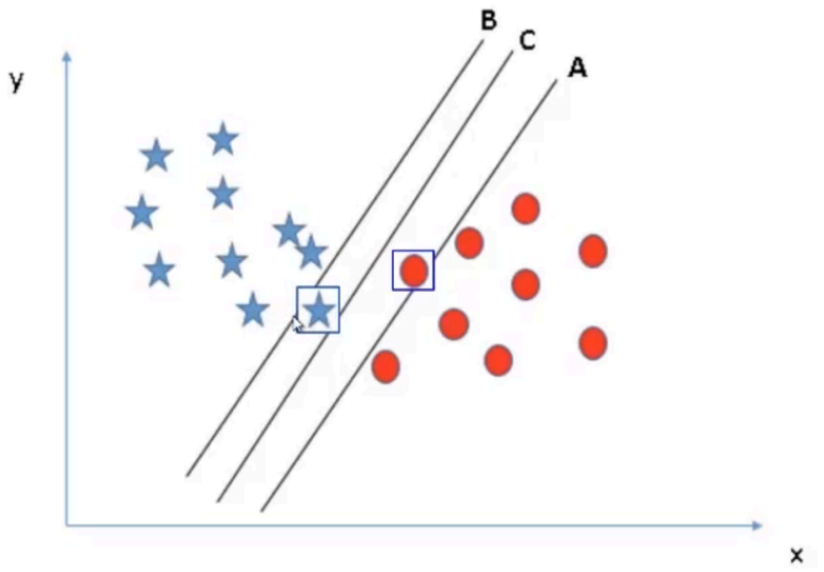


Figure 2. Visual representation of support vector machines.

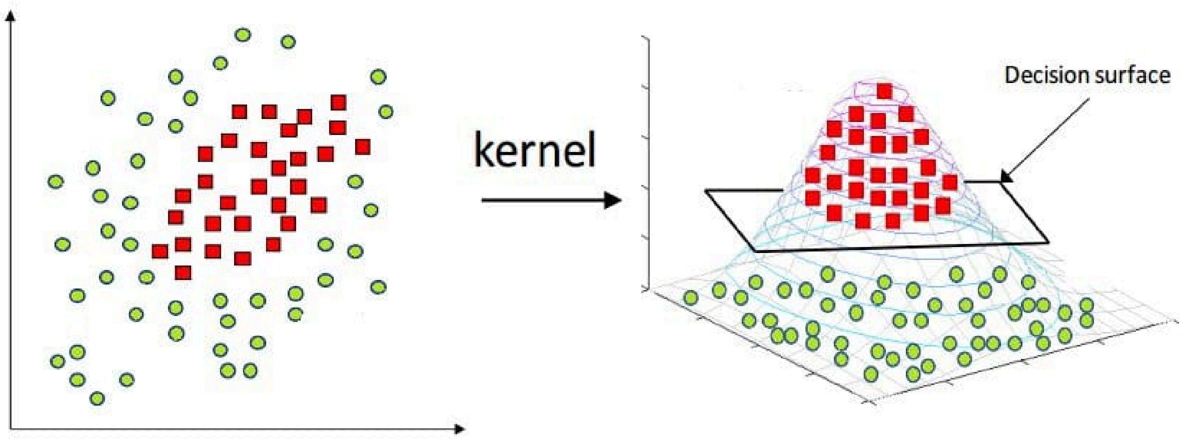


Figure 3. Kernel trick explained visually.

2.1.2 Convolutional Neural Network

[9] is one of the most revolutionary works in the fields of Computer Science and Deep Learning, presenting Convolutional Neural Networks (CNNs). This structure changed the approach to image classification tasks completely and made a significant impact to the whole world of deep learning and artificial intelligence that was only gaining momentum at that time. CNNs make the use of small NxN kernels for dimensionality reduction and feature map pooling and extraction, which made all image processing related tasks faster and allowed to go deeper into learning, due to the number of parameters per layer was significantly decreased. [9] also introduced the first CNN based model(Figure 4), called LeNet.

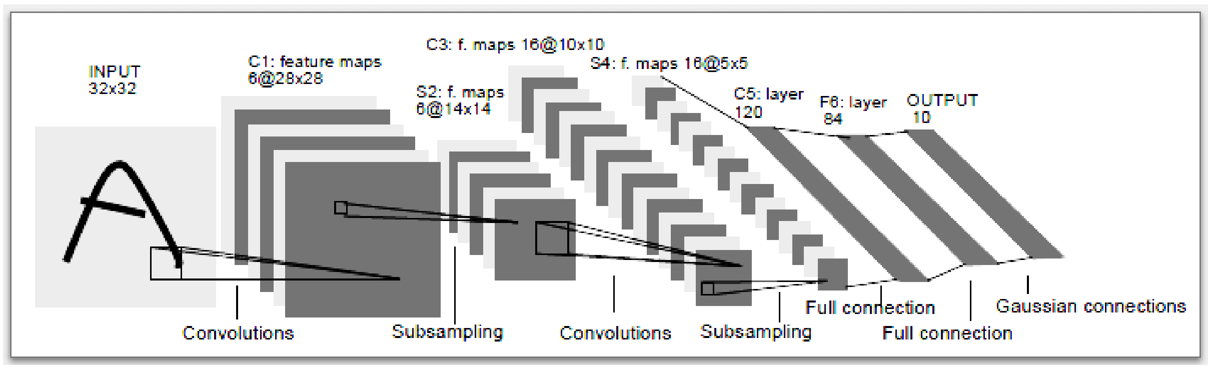


Figure 4. LeNet architecture that introduced the use of CNNs in [9]

2.1.3 ResNet

Residual Network(ResNet), which was introduced by [10], is considered to be a state-of-the-art(SOTA) and classic model in image classification. The main concept behind this architecture uses residual blocks(Figure 5), which helps with mapping features and improves the accuracy and speed of the model while having fewer parameters. Figure 5 shows the skip-connection, a shortcut, that implements the y = F(x) + x equation. Skip-connections address the issue of vanishing or degraded gradients. Figure 6 shows the architecture for ResNet-34, a 34-layered version, demonstrating shortcuts implemented on a diagram(dotted connections increase dimensions).

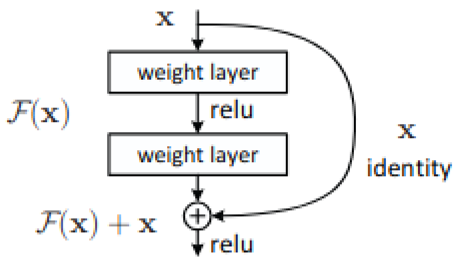

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 5. Residual Block scheme

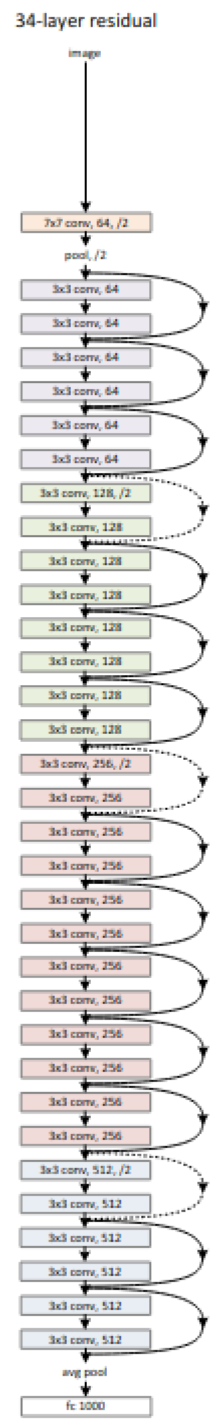


Figure 6. ResNet-34 architecture by [10]

ResNet is widely used for image classification [10], segmentation [11][12] and object detection [13] either directly or as a backbone.

2.2 Quantum Computing solutions

2.2.1 Quantum Support Vector Machines

Classic Support Vector Machines have the time complexity of *O(n³)*, due to introduced kernel methods [16][17] that deal with such issues as non-linear training data. [14] showed that a time complexity can be improved to *O(log M N)* by using quantum random memory access(qRAM). In order to do this, a quantum kernel was introduced. In general the quantum kernel itself has the following tracing:

$$\mathrm{tr2}\{|X><X|\} = \frac{1}{N_X}\sum_{i,j=1}^{M} < \overline{x_i}|\overline{x_j} > |\overline{x_i}||\overline{x_j}||i >< j| \; = \; \frac{K}{trK}$$

$$\text{where } |X> = 1/\sqrt{N_x}\sum_{i=1}^{M} |\overline{x_i}||i > |\overline{x_i} >$$

However, in order to apply it to supervised learning using quantum computing, the inverse of $\widehat{K}^{-1}$ needs to be able to enact $e^{-i\widehat{K}\Delta t}$ efficiently (Rebentrost et al., 2014). To address this issue, considering the kernel is non-sparse, an algorithm in [15] was applied, giving the following formulation:

$$e^{-iL_{\widehat{K}}\Delta t}(\rho) \; \approx \; tr_1\{e^{-iS\Delta t}\widehat{K} \otimes \rho e^{iS\Delta t}\} \; = \; \rho - i\Delta t|\widehat{K}, \rho| + O(\Delta t^2)$$

$$\text{where } S \; = \; \sum_{m,n=1}^{M} |m >< n| \otimes |n >< m|$$

Such kernels are applicable to a wide range of quantum machine learning algorithms, including image classification. [19] used Quantum Support Vector Machines (QSVM) for classifying brain tumors based on MRI scan results.

2.2.2 Quantum Convolutional Neural Networks

Inspired by the architecture of CNNs [18] designed a similar structure that extends CNNs features into the field of quantum computing and machine learning. In Quantum Convolutional Neural Network (QCNN) (Figure 7), a single quasi-local unitary gate U_i is applied to input quantum state ρ_{in} in a transactionally-invariant manner for finite depth (Cong et al., 2019). As for pooling, in [18] a fraction of qubits is measured, which then determines the V_j rotations for other nearby qubits. Convolutions and pooling are repeated for a certain amount of times to ensure sufficient dimensionality reduction, which is then followed by another unitary F that serves as a fully-connected layer at the end, the output is measuring a certain amount of qubits.

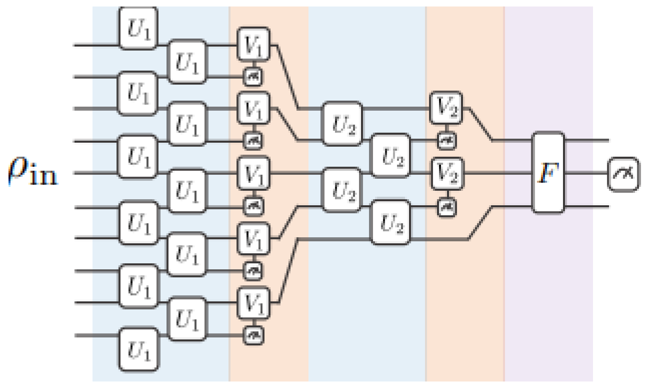


Figure 7. QCNN architecture.

2.3 Variational Quantum Circuit

Variational Quantum Circuit(VQC) is a hybrid quantum-classic algorithm for quantum machine learning tasks. The main principle is using quantum circuits with adjustable parameters. VQC consists of the following three stages:

- Encoding of classical data to quantum state
- Application of parameterised rotations and entanglements
- Measuring of qubits

Proposed methodology

3.1 Data preparation and processing pipeline

The data used for the task of image classification is taken from Kaggle platform [21]. The dataset has three splits: train, val and test. The train is the biggest one and imbalanced, since there are only 1341 cases of healthy patients(no pneumonia), while diagnosed pneumonia cases count in the train directory is 3875. Hence, the amount of pneumonia patients was randomly capped at 1341(random under sampling technique applied). It may potentially impact the accuracy, but in case a hybrid model performs quite well on capped dataset, it may as well perform better with full dataset(with techniques dealing with class imbalances applied, of course).
A standard pipeline was applied, all images are resized to 224x224 size and normalized. Cases of healthy patients are labelled as ‘NORMAL’, mapped to 0, and pneumonia cases are labelled as ‘PNEUMONIA’, mapped to 1.

3.2 Hybrid model architecture

This paper presents a hybrid solution, meaning that both classic machine learning principles will be combined with quantum solutions. For the classic part a ResNet [10] with 50 layers pre-trained on ImageNet dataset [20] is used and for quantum layer a basic VQC is implemented and added on top of classic fully-connected layer of ResNet-50. The original ResNet backbone has a fully-connected layer predicting 1000 classes, which is changed in this instance to two outputs, which are then converted to qubits for the VQC gate, which consist of 4 quantum layers each processing 2 qubits - each qubit is rotated in 3 directions - x, y and z. Output qubits are processed through a Pauli-Z gate and measured to get classic bits for the softmax activation layer.

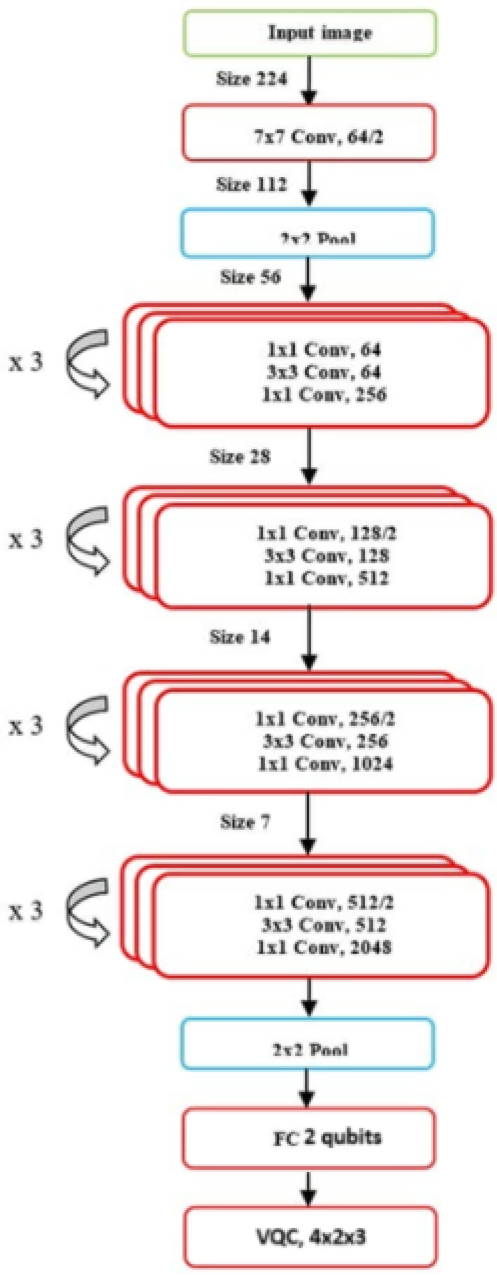


Figure 8. ResNet and VQC hybrid architecture

3.3 Training parameters

The whole model was trained for only 10 epochs with a learning rate of 0.1. The optimization algorithm used is a classic Stochastic Gradient Descent(SGD) as this solver goes very well for the ResNet fine-tuning process, since it initially was trained using SGD as its optimizer [10]. The batch size is 16. Given more data and a stronger system set up, the amount of epochs might be increased. In such a case, a learning rate scheduler might be applied -

dividing learning rate by 10 every 10 steps. The loss function is CrossEntropyLoss. For full architecture implementation, training results, code and dataset - see Appendix A.

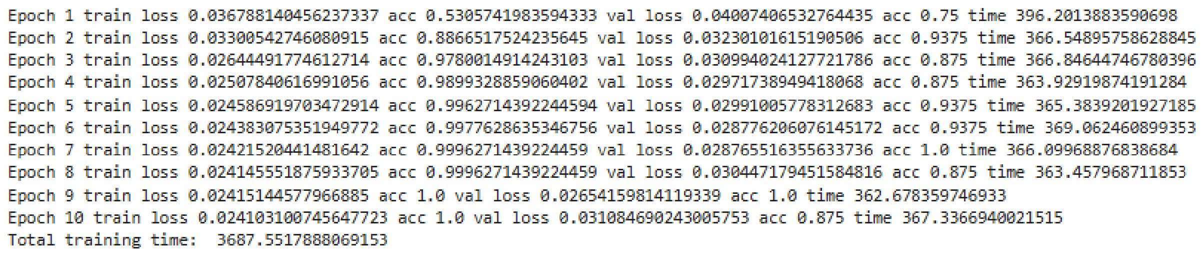
Experiment results evaluation

4.1 Training process log results analysis

This section introduces loss and accuracy logs for both training and validation sets for hybrid and classic versions. Classic ResNet50 was fine-tuned using the same training settings as discussed in chapter 3, section 3.3

4.1.1 Hybrid model outputs

Figures 9 and 10 demonstrate fine-tuning process log displaying average training and validation losses and accuracies after each epoch and total time required to complete each epochs and total time it took for all 10 epochs to run.



```
Epoch 1 train loss 0.036788140456237337 acc 0.5305741983594333 val loss 0.04007406532764435 acc 0.75 time 396.2013883590698
Epoch 2 train loss 0.03300542746080915 acc 0.8866517524235645 val loss 0.03230101615190506 acc 0.9375 time 366.54895758628845
Epoch 3 train loss 0.02644491774612714 acc 0.9780014914243103 val loss 0.030994024127721786 acc 0.875 time 366.84644746780396
Epoch 4 train loss 0.02507840616991056 acc 0.9899328859060402 val loss 0.02971738949418068 acc 0.875 time 363.92919874191284
Epoch 5 train loss 0.024586919703472914 acc 0.9962714392244594 val loss 0.02991005778312683 acc 0.9375 time 365.3839201927185
Epoch 6 train loss 0.024383075351949772 acc 0.9977628635346756 val loss 0.028776206076145172 acc 0.9375 time 369.062460899353
Epoch 7 train loss 0.02421520441481642 acc 0.9996271439224459 val loss 0.028765516355633736 acc 1.0 time 366.09968876838684
Epoch 8 train loss 0.024145551875933705 acc 0.9996271439224459 val loss 0.030447179451584816 acc 0.875 time 363.457968711853
Epoch 9 train loss 0.02415144577966885 acc 1.0 val loss 0.02654159814119339 acc 1.0 time 362.678359746933
Epoch 10 train loss 0.02410310074564772 3 acc 1.0 val loss 0.031084690243005753 acc 0.875 time 367.3366940021515
Total training time:  3687.5517888069153
```

Figure 9. Fine-tuning process log metrics recorded in detail

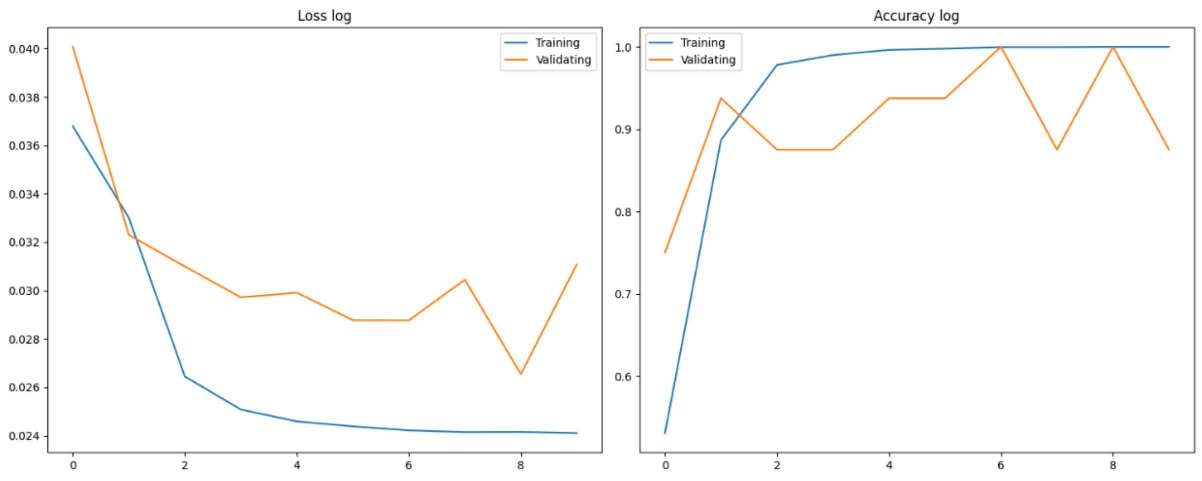
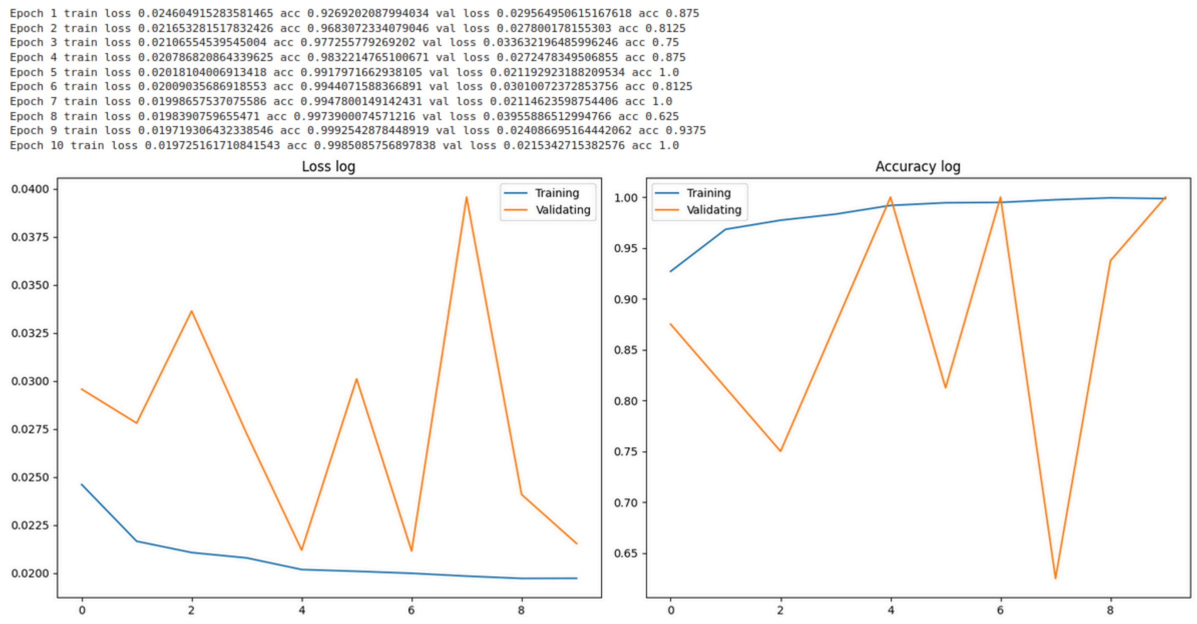


Figure 10. Line plots showing the dynamics between training and validation losses and accuracies throughout the fine-tuning process

Training loss and accuracy records progress smoothly, loss - steadily going lower, accuracy - going higher. Validation loss shows fluctuations after epoch 6 and fluctuations during accuracy records. This might be the sign of slight overfitting - when a model performs well during training and bad on validation sets. That might be related to a relatively small dataset and very basic training set up. Potential improvements would be adding dropout to a fully-connected layer, tuning optimization parameters, since the only argument to be passed is the learning rate.

4.1.2 Classic ResNet50 outputs



```
Epoch 1 train loss 0.024604915283581465 acc 0.9269202087994034 val loss 0.029564950615167618 acc 0.875
Epoch 2 train loss 0.021653281517832426 acc 0.9683072334079046 val loss 0.027800178155303 acc 0.8125
Epoch 3 train loss 0.02106554539545004 acc 0.977255779269202 val loss 0.033632196485996246 acc 0.75
Epoch 4 train loss 0.020786820864339625 acc 0.9832214765100671 val loss 0.0272478349506855 acc 0.875
Epoch 5 train loss 0.02018104006913418 acc 0.9917971662938105 val loss 0.021192923188209534 acc 1.0
Epoch 6 train loss 0.02009035686918553 acc 0.9944071588366891 val loss 0.03010072372853756 acc 0.8125
Epoch 7 train loss 0.01998657537075586 acc 0.9947800149142431 val loss 0.02114623598754406 acc 1.0
Epoch 8 train loss 0.0198390759655471 acc 0.9973900074571216 val loss 0.03955886512994766 acc 0.625
Epoch 9 train loss 0.019719306432338546 acc 0.9992542878448919 val loss 0.024086695164442062 acc 0.9375
Epoch 10 train loss 0.019725161710841543 acc 0.9985085756897838 val loss 0.0215342715382576 acc 1.0
```

Figure 11. Classic ResNet50 fine-tuning logs

As Figure 11 demonstrated, training losses and accuracy scores were improving smoothly, but with very slight changes, since both of them started strong initially. However, for validation sets, loss and accuracy scores were fluctuating severely throughout the whole fine-tuning process.

4.2 Testing fine-tuned model: metric results

This section presents metric results of models that were already fine-tuned.

4.2.1 Hybrid model outputs

```
              precision    recall  f1-score   support

           0       0.70      0.99      0.82       165
           1       1.00      0.78      0.88       320

    accuracy                           0.85       485
   macro avg       0.85      0.89      0.85       485
weighted avg       0.90      0.85      0.86       485
```

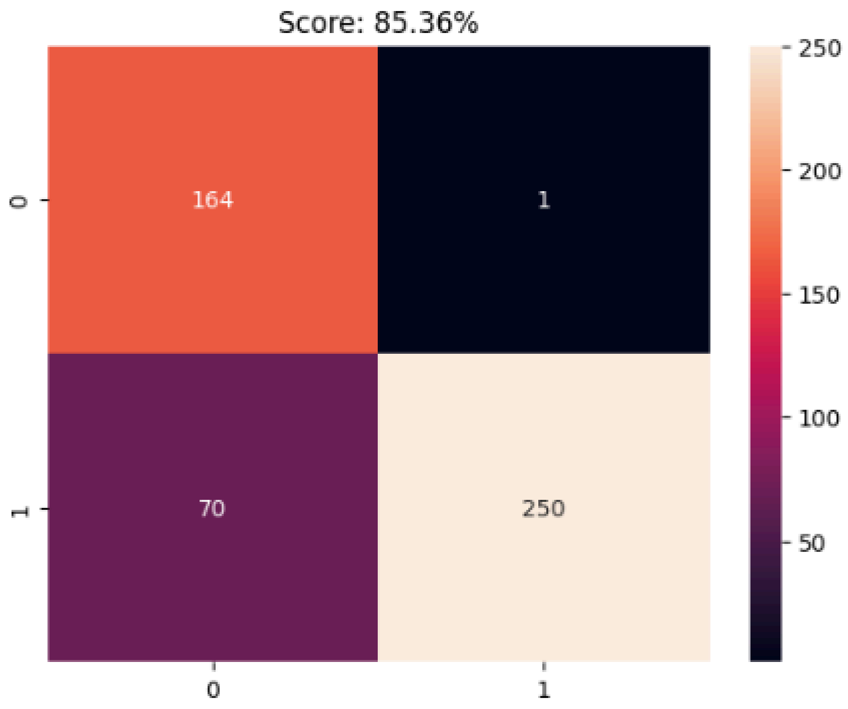


Figure 11. Fine-tuned model evaluation results.

Figure 11 demonstrates that there is a slight bias towards predicting that a patient has pneumonia as far as the precision for both classes is considered. But in general overall accuracy score, f1 score for both classes is on a good level.

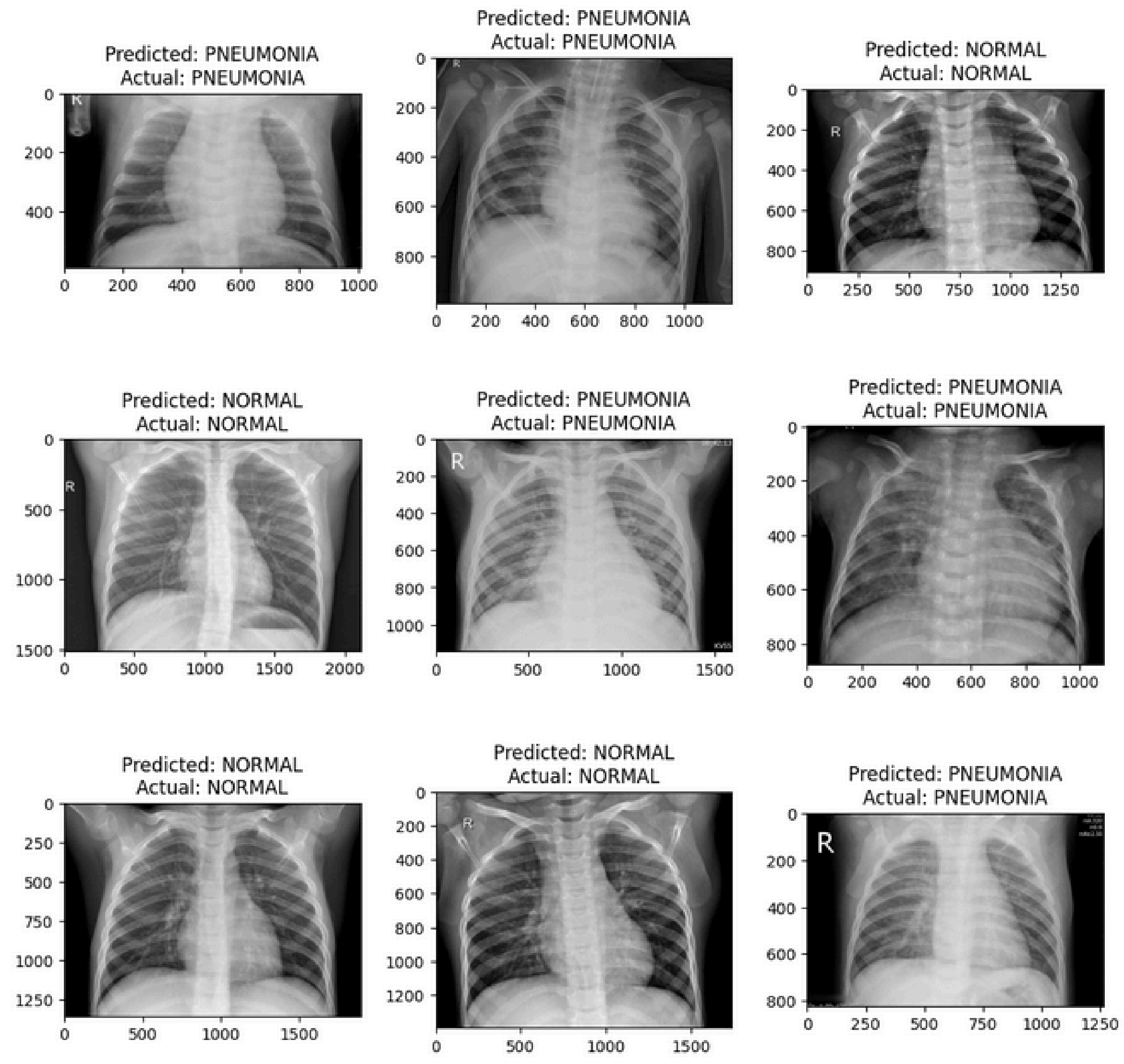


Figure 12. Inference of hybrid model after fine-tuning.

4.2.2 Classic model outputs

```
              precision    recall  f1-score   support

           0       0.79      0.99      0.88       187
           1       0.99      0.84      0.91       298

    accuracy                           0.89       485
   macro avg       0.89      0.91      0.89       485
weighted avg       0.91      0.89      0.90       485
```

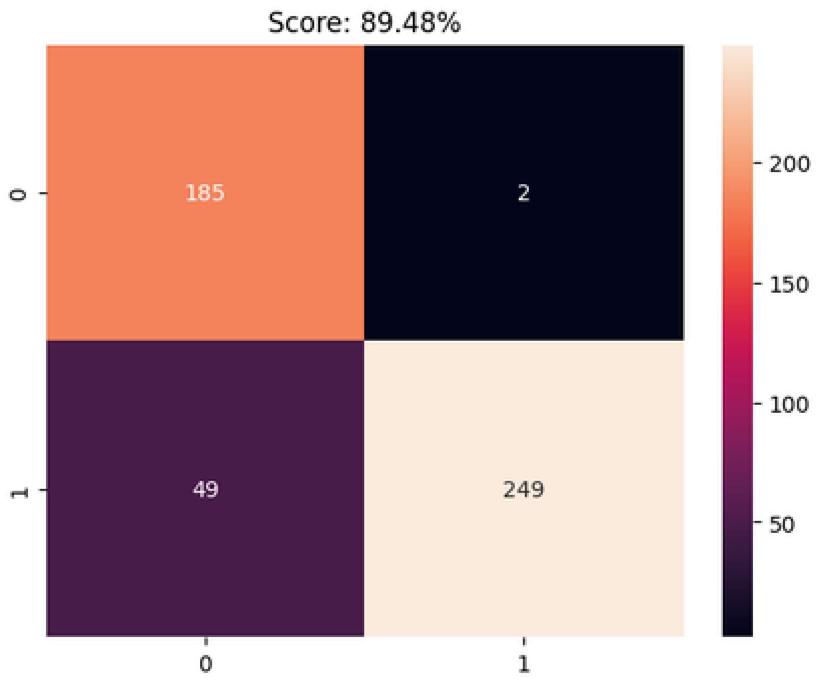


Figure 13. ResNet50 classic model evaluation results

Figure 12 shows that classic ResNet50 has better accuracy and precision results: smaller precision gap(still hinting towards pneumonia patients, when predicting), higher overall accuracy. This further indicates a classic well established method would output better results than a fusion of quantum and classic methods.

4.3 Experiment results conclusions

ResNet50 fused with VQC demonstrated steadier progress during fine-tuning in the evaluation phase, such as steadier decrease in loss and steadier growth of accuracy scores, with smaller fluctuation rates. Classic ResNet showed higher fluctuations rate for validation phase. Both solutions showed steady progress, such as smooth training loss decrement and steady accuracy score increment, which is hinting towards slight over-fitting. However, when testing both models after fine-tuning, both performed relatively good. Testing showed the classic version having the upper hand in evaluation metrics: smaller difference gaps between labels precision scores, higher f1 scores for each predicted class, and overall accuracy score is higher by 4%.

The fine-tuning process for the classic version was completed way faster than for the hybrid one, mostly due to additional process and tuning of quantum parameters in rotational gates.

However, this experiment showed that with better hardware, balanced dataset and advanced training methods hybrid models have the potential to rival classic models. This might indicate the true potential of quantum machine learning.

Conclusion

Quantum computing developed when physicists started applying principles of quantum physics to classic computing algorithms. Such quantum algorithms as Shor's algorithms for large-number factorisation or Grover's searching algorithm for unstructured systems demonstrated time complexity improvement and optimisation prospects. This paper researches effects of fusing quantum and classic solutions for the task of image classification by solving the problem of detecting pneumonia patients on chest X-Ray scans. The main architecture discussed is Variational Quantum Circuit added on top of a pre-trained ResNet50. Hybrid and classic models were trained to detect pneumonia in similar training settings and parameters. The training process showed classic models to have more fluctuations in validating the model, while hybrid models demonstrated smoother progression. After running the model on the test data, the classic model showed a slight upper hand in performance metrics over the hybrid version. That only showed that hybrid variants and, potentially, fully quantum solutions might rival classic ones, given that quantum and hybrid variants only use simulation on classic hardware.

References:

[1] DPMA | Jack Kilby & Integrated Circuit. (2024, January 31). Deutsches Patent- Und Markenamt. https://www.dpma.de/english/our_office/publications/milestones/computerpioneers/65yearsintegratedcircuit/index.html
[2] Moore, G. E. (1998). Cramming More Components onto Integrated Circuits. Proceedings of the IEEE, 86(1), 82–85. https://doi.org/10.1109/jproc.1998.658762
[3] Du, Y., Wang, X., Guo, N., Yu, Z., Qian, Y., Zhang, K., Hsieh, M.-H., Rebentrost, P., & Tao, D. (2025). *Quantum Machine Learning: A Hands-on Tutorial for Machine Learning Practitioners and Researchers*. ArXiv.org. https://arxiv.org/abs/2502.01146
[4] Wikipedia Contributors. (2019, March 27). *Quantum computing*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Quantum_computing
[5] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*. ArXiv.org; arXiv. https://arxiv.org/abs/1512.03385
[6] Shukla, iam@Tanmay. (2024). *Pneumonia Dataset with Chest X-Ray*. Kaggle.com. https://www.kaggle.com/datasets/iamtanmayshukla/pneumonia-radiography-dataset/data
[7] Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Springer New York. https://doi.org/10.1007/978-1-4757-3264-1
[8] Chandra, M. A., & Bedi, S. S. (2018). Survey on SVM and their application in image classification. *International Journal of Information Technology*, *13*. https://doi.org/10.1007/s41870-017-0080-1
[9] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

[10] He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*. ArXiv.org; arXiv. https://arxiv.org/abs/1512.03385
[11] Sahragard, E., Farsi, H., & Mohamadzadeh, S. (2025). Semantic Segmentation Using an Improved ResNet Structure and Efficient Channel Attention Mechanism Applied to Atrous Spatial Pyramid Pooling in a Fully Convolutional Network. *International Journal of Engineering*, *38*(11), 2511–2526. https://doi.org/10.5829/ije.2025.38.11b.04
[12] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *ArXiv:1606.00915 [Cs]*. https://arxiv.org/abs/1606.00915
[13] Lu, X., Kang, X., Nishide, S., & Ren, F. (2019). Object detection based on SSD-ResNet. *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. https://doi.org/10.1109/ccis48116.2019.9073753
[14] Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, *113*(13). https://doi.org/10.1103/physrevlett.113.130503
[15] Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum principal component analysis. *Nature Physics*, *10*(9), 631–633. https://doi.org/10.1038/nphys3029
[16] Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171–1220. https://doi.org/10.1214/009053607000000677
[17] Muller, K.-R. ., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*(2), 181–201. https://doi.org/10.1109/72.914517
[18] Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. *Nature Physics*, *15*(12), 1273–1278. https://doi.org/10.1038/s41567-019-0648-8
[19] Kumar, T., Kumar, D., & Singh, G. (2023). Brain Tumour Classification Using Quantum Support Vector Machine Learning Algorithm. *IETE Journal of Research*, 1–14. https://doi.org/10.1080/03772063.2023.2245350
[20] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/cvpr.2009.5206848
[21] Shukla, T. (2024). *Pneumonia Dataset with Chest X-Ray*. Kaggle.com. https://www.kaggle.com/datasets/iamtanmayshukla/pneumonia-radiography-dataset

Appendix A:

- Dataset link: https://www.kaggle.com/datasets/iamtanmayshukla/pneumonia-radiography-dataset
- GitHub repository link with solutions and training results on Jupyter notebooks: https://github.com/a3a256/ResNet-VQC