An Investigation Into Quantum Machine Learning Based Image Classification
Northumbria University

Philip Easom-McCaldin

November 2021

Abstract

Image classification is dominated by high-performing deep learning methods, that typically take on the form of large-scale convolutional neural networks. Over the past decade, the magnitude of these systems has grown to the point where a moderately-sized system can consist of millions of trainable parameters. The subsequent impact on the computational load during optimization and inference is massive, therefore a change in approach is needed to reduce the scale of this problem. Quantum computing is a modern development that utilises natural quantum-mechanical principles, in an effort to efficiently process data in a classically-intractable manner. Furthermore, quantum machine learning, the application of quantum computing for machine learning tasks, has seen a surge in interest and development. This makes quantum computing an appealing area of research to consider for solutions to the problems faced. Firstly, this thesis investigates the classical-based solution of transfer learning to reduce the computational load of optimizing deep learning algorithms. Following this, quantum-based methods are analysed to determine their effectiveness for the task of image classification. This culminates in the proposal of a novel quantum image classification algorithm. This thesis makes several contributions of knowledge to the working area. Firstly, it is demonstrated that a computational speedup can be gained via quantum routines over classical algorithms for image classification. However, if a classical approach is preferred, then it is presented that transfer learning can maintain classification performance whilst negating the costly optimization of a large proportion of parameters. Secondly, an enhanced understanding of single-qubit encoding is gained. Experimental results show that substantial classification accuracy improvements can be made as data encodings are repeated. In addition, results support that an element of robustness to environmental noise can be gained for repeated encodings, which is important to consider in the NISQ era. Finally, a novel quantum image classification algorithm is proposed, which demonstrates that a lone qubit is a capable image classifier. Results determine that classification accuracies in the 90th percentile can be achieved using a minimum of 6 working parameters. Overall, this research may have a large impact towards the development of quantum image classification algorithms, where a plethora of options for future development are opened as well.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to give thanks to Richard Jiang, for initially providing me with the opportunity to pursue a PhD in the first place, as well as for continual support even after departing from the university. Secondly, I would like to thank the rest of my supervision team, consisting (in order of meeting you) of Ahmed Bouridane and Ammar Belatreche. Although some challenges were faced along the course of this PhD, support and understanding was always given where required. Your valuable insight and feedback to my work was greatly appreciated, and helped me to develop my own abilities to a great extent.

Finally, I would like to dedicate this thesis to my wife Paige and 3-month old daughter Poppy. The past three years have not necessarily been easy, however your support and listening-ear over this time has helped me to strive to better myself and get the most out of life. I will always be grateful to you for doing this, thank you.

Declaration

I declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others. Code in order to reproduce results contained in this thesis is also available upon request to philip.easom@northumbria.ac.uk.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the University Ethics Committee on 25/07/2019 (ref: 17602).

I declare that the word count of this Thesis is 43,214 words.

Name: Philip Easom-McCaldin

Date: 30/11/2021

Chapter 1

Introduction

1.1 Project Background

Image classification is a fundamental area of research within the modern day field of machine learning (ML). The typical approach to image classification tasks include a feature extraction step of some sort, prior to inputting these features into a classifier and optimization against a cost function [1]. Currently, the most popular approach is to use deep learning (DL) architectures [2], which aim to combine the steps of feature extraction and classification into a single, trainable architecture. Many DL architectures fall into the category of convolutional neural networks (CNNs), which utilize a mass of kernel operations congested into a series of layers, that output a set of extracted features for classification using a traditional 'fully-connected' feed-forward neural network. These network architectures have shown to be extremely successful at tasks they are trained to, often dubbed at beating human-level performance [3] at the same time.

Interest and research development towards CNNs have seen a tremendous amount of growth within the past decade alone. Improvements to the processing capability of graphical processing unit (GPU) systems has had a direct correlation with the subsequent depth and complexity of CNN architectures. The ability to compute large volumes of data with modern GPUs has enabled DL and computer vision (CV) research to be conducted at a rapid rate. However, not only are classification performance improvements generally small, they often come at the expense of the introduction of vast webs of interconnecting convolutional layers and filters that increase the algorithm complexity and parameter count somewhat.

To put this statement into perspective, consider one of the earliest CNN architectures, LeNet5 [4]. The LeNet5 architecture had 7 layers total, consisting of 2 sets of convolutional and pooling layers to extract features and reduce dimensionality, followed by a flattening convolutional layer which was then proceeded by a string of 2 fully-connected layers and a final softmax classifier. This basic, but powerful architecture was able to achieve a test-set accuracy over 99.1% on the MNIST handwritten digits dataset [5], using $\sim 60,000$ trainable parameters. However, in contrast to this, one of the highest performing state-of-the-art architectures [6] could reach an extremely strong test-set accuracy of 99.83%, yet required optimization of $\sim$ 1.4 million parameters. This equates to over 23 times the amount of parameters that LeNet5 required, for a $< 1\%$ classification accuracy improvement.

Whilst clearly reaching strong classification accuracy levels is a positive aspect for computer vision purposes, is gaining that extra small 1-2% of accuracy always worth the increase in computational overhead and components that need optimizing? Whilst some could argue that this tradeoff may be necessary for certain applications, others may disagree by saying that CNN methodologies are stagnating and that we should be focusing our efforts elsewhere, rather than throw more GPU computing power at the training of algorithms. Nevertheless, it is recognised that developments to GPU hardware will not be able to continue at this rate of growth indefinitely, therefore we will be required to search for alternative solutions at some point or another.

The downside of large-scale parametrization for optimization and computation time has been considered to some extent during this period of DL development. An alternative method attempting to reduce optimization time has taken advantage of the copious amounts of raw data provided by large-scale datasets such as the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) dataset [7]. There exists CCN architectures which have taken advantage of this readily-available data and trained for a large amount of time using it. The trained weight values are then taken and saved to an open-access repository or library interface for instant further use, without having to repeat the full training process. Further developments or applications have then taken this 'pre-trained' network, prior to task-specific training on a specialized dataset of some kind. This practice has become known as *transfer learning* [8].

In practical terms, during transfer learning the feature extraction portion of the CNN is left frozen, where only weights contained within the final few layers are considered for optimization. Whilst this practice still utilizes the entirety of the DL network for the forward pass, it can reduce computational load by requiring the calculation of fewer gradients in total during backpropagation. However, it is still unclear to what extent transfer learning can maintain classification performance, whilst reducing computational load during optimization. It is therefore relevant to explore the aspect of whether transfer learning can be a temporary effective solution to the problems described, or whether alternative methods of ML algorithm design should be investigated.

Quantum computing (QC) is a method of computing that has seen a dramatic growth of interest within recent years. Based on quantum-mechanical principles of superposition and entanglement, arguably no other field has seen as much uninformed speculation as this has, not when quantum computing for ML is considered. Whilst the nature of quantum information does possess an innate parallelism, it can be easy to fall for the narrative that quantum machine learning (QML) will be the primary solution to all ML tasks, especially when exponential computational complexity reductions are often promised. But with that said, it does not mean that QML cannot still be effective at providing a solution to certain ML tasks, it is simply another computational tool whose potential is yet to be realized.

With an array of quantum computing based algorithms proposed to date, [9–12] to name a few, there is a direct need to explore how QC can be adopted, modified and applied to form a suitable, image-based classification tool. By doing so, this research aims to create a set of robust, foundational quantum classifiers that present advantages over classical algorithms, whilst giving ample opportunities for further investigation and development. It is hoped that by doing so, we can develop QML algorithms that can improve performance to given ML tasks over other classical methodologies, as well as improve our working knowledge of how QML algorithms operate in practice.

1.2 Research Aim & Questions

Following the discussion of certain problems facing image classification and the interest towards quantum computing, the subsequent direction and aims for this thesis should be proposed. The aim of this thesis as a whole is to **investigate the transition between classical and quantum computing-based image classification, in relation to how quantum machine learning may enhance classification performance or reduce computational complexity**. In order to reach this aim, the following research questions and sub-objectives are proposed:

Research Question 1 *To what extent can the application of transfer learning maintain classification performance of deep CNNs, whilst reducing computational overhead from the optimization of large parameter counts?*

Research Question 2 *To what extent can a non-gradient based approach be taken to perform effective quantum image classification?*

Research Question 3 *Can effective feature representations of classical data be created in Hilbert space using low numbers of quantum bits?*

Research Question 4 *Can a novel quantum machine learning classifier be developed for image classification that makes efficient use of resources provided?*

1.3 Thesis Synopsis & Structure

In order to answer the research questions outlined in section 1.2, this thesis will be organized in the following format:

Chapter 2 provides a systematic introduction of working background knowledge for classical and quantum computing terminologies, to provide an understanding for future chapters. This chapter will begin by introducing classical deep learning through feed-forward neural networks, as well as convolutional neural networks. Following this, an overview of quantum computing will be showcased in order to provide the knowledge required to understand fundamental differences between classical and quantum computation. This aspect of the chapter will begin with the introduction of the quantum bit, shortly followed by quantum information representation, how quantum operations are performed and the application of quantum computing for machine learning.

Chapter 3 contains a discussion and review of relevant literature surrounding the field of quantum machine learning. Firstly, an overview of the field is described, which sets the area and boundaries for the review. Following this, the discussion starts with quantum-based algorithms and solutions to regression-based tasks. Then, the review extends to quantum-based classification proposals. Here, various approaches are considered, such as quantum support vector machine proposals, quantum k-nearest neighbour algorithms and other proposed methods that rely on a backbone of variational quantum circuits.

Chapter 4 presents the first working chapter of the thesis, which focuses towards answering research question 1. This chapter contains a review of state of the art CNNs, tied into both image classification and object recognition tasks. The application of transfer learning is conducted to a specific task using the deep learning methodology YOLOv2. Performance results are then analysed, where a discussion on the effectiveness of transfer learning as a solution to reducing computation is provided.

Chapter 5 presents an application of quantum computing towards a facial identification classification task with the intent of answering research question 2. The methodology presented within this chapter utilises fidelity-based measurements of similarity combined with a threshold, rather than gradient based methods of optimization which can be time-consuming. An analysis of the methodologies performance is given with respect to various classical ML algorithms, as well as a discussion in relation to the field as a whole.

Chapter 6 provides a review of quantum information encoding methods, and showcases an in-depth analysis of the 'data reuploading' methodology, with intent of answering research question 3. Here, results are obtained that analyse the method over numerous classification datasets, whilst remaining in context of the NISQ era of quantum computing.

Chapter 7 is the final working content chapter of the thesis, with the aim of completing the final research objective (research question 4). In this chapter a novel quantum image classification methodology is presented, that is based on single-qubit encoding with the advantage of maintaining spatial relationships between pixels. Experimental results are gathered using multiple datasets, where an analysis is provided to evaluate the working effectiveness of the proposed methodology for image classification in the NISQ era.

Chapter 8 summarises the contributions made in this thesis, including a discussion of potential future works.

1.4 Publications Made

As a statement of authorship, excerpts of this thesis have been published in the following peer-reviewed articles as part of the project progression and development between 2018 - 2021, in order of appearence within this thesis:

1. P. Easom, A. Bouridane, F. Qiang, L. Zhang, C. Downs and R. Jiang, *"In-House Deep Environmental Sentience for Smart Homecare Solutions Toward Ageing Society,"* 2020 International Conference on Machine Learning and Cybernetics (ICMLC), 2020, pp. 261-266, doi: 10.1109/ICMLC51923.2020.9469531.

2. P. Easom-McCaldin, A. Bouridane, A. Belatreche and R. Jiang, 2020. *"Towards building a facial identification system using quantum machine learning techniques."* Accepted Aug 2020 for publishment in Journal of Advances in Information Technology (JAIT).

3. P. Easom-McCaldin, A. Bouridane, A. Belatreche and R. Jiang, *"On Depth, Robustness and Performance Using the Data Re-Uploading Single-Qubit Classifier,"* in IEEE Access, vol. 9, pp. 65127-65139, 2021, doi: 10.1109/ACCESS.2021.3075492.

4. P. Easom-McCaldin, A. Bouridane, A. Belatreche, R. Jiang and S. Al-Maadeed, 2021. *"Efficient Quantum Image Classification Using Single Qubit Encoding."* Submitted Oct 2021 to IEEE Transactions on Neural Networks and Learning Systems (TNNLS).

1.5 Thesis Contributions

To summarise the content in this thesis, the key contributions made through the research contained within chapters 3 to 6 are, in order of appearance:

- Transfer learning remains a viable technique to consider if the reduction of computational load during optimization is a priority. Experimental results demonstrated that a good classification accuracy result of 91.47% was achieved, whilst removing the optimization requirement for 76.9% of the total parameterized layers. This is presented throughout research within chapter 3 and in publication number 1 in section 1.4.

- A fidelity-based approach is capable of effective classification, whilst removing the costly process of backpropagation for quantum-based image classification. Notably, the implemented method improved computational complexity over similar classical algorithms through utilising resource-efficient amplitude encoding. This is presented throughout research contained within chapter 5 and in publication number 2 in section 1.4.

- A novel quantum image classifier was proposed based on the single-qubit encoding scheme. Initial experimental results for the encoding scheme showed a promising capability to classify complex datasets. The greatest performance improvements were also determined to occur within 1-3 data encoding layers. The proposed algorithm modified this method in order to maintain pixel relationships, which greatly reduced the parameter count with itent of shortening computational load during optimization. Experimental results determined that the proposed method had great potential as a quantum image classifier, whilst remaining efficient in its use of quantum resources and robust to certain environmental noise impacts. This finding is presented over research contained within chapters 5 and 6, as well as within publications 3 and 4 listed in section 1.4.

Chapter 2

Background & Literature Review

In this chapter, the topics of DL and QC are introduced to provide a foundational level of knowledge required to understand concepts that are used throughout this thesis. The classical-computing aspect of the chapter begins by defining neural network methods, leading up to the introduction of CNNs. In contrast, the quantum-computing aspect of the chapter begins with defining how quantum information is represented, followed by quantum-based operations and various quantum-measurement protocols. In addition to this, reviews of relevant literature are discussed for both CNN methodologies contained within DL, as well as various applications of QC for ML purposes. This is done to provide a basis for the research that is contained within this thesis.

2.1 Introduction to Deep Learning

2.1.1 Feed-Forward Neural Networks

In order to introduce DL, it is convenient to begin at the most basic form of neural network that provides the basis for further explanation, the *perceptron.* A percepton is a supervised linear binary classifier that uses a row vector of inputs $\mathbf{x} = [x_1, x_2, ..., x_N] \in \mathbb{R}^N$, associated weight values $\mathbf{w} = [w_1, w_2, ..., w_N] \in \mathbb{R}^N$, and a bias term $b \in \mathbb{R}$ to produce an output $\hat{y}$. In mathematical terms, the output $\hat{y}$ can be defined as:

$$\hat{y} \;=\; \mathbf{x}^T\mathbf{w} + b \;=\; \sum_{i=1}^{N} x_i w_i + b \tag{2.1}$$

The output value $\hat{y}$ is then fed through a form of activation function to map the input x to an output value $f(x)$. For a perceptron, a unit-step function is used to produce a single binary output value, where:

$$f(x) = \begin{cases} 1 & \text{if } \mathbf{x}^T\mathbf{w} + b \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{2.2}$$

The entire process described above of the perceptron can subsequently be illustrated in figure 2.1 as:

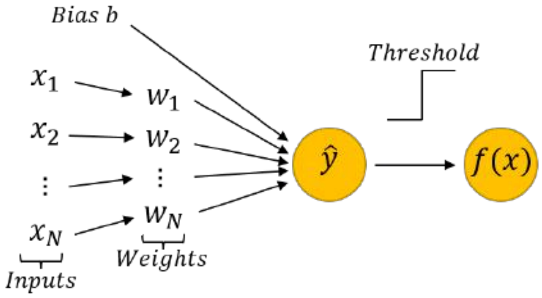


Figure 2.1: The flow of input to output for a perceptron.

Whilst the perceptron is capable of classifying datasets that are linearly seperable in nature, they are severley limited by datasets that do not comply with this and are non-linear by nature. In order to overcome these limitations that single perceptrons are susceptible to, multiple perceptrons can be stacked together to form a series of layers capable of classifying more complex forms of data. The resulting neural network formed by stacking perceptrons together is known as a *multilayer perceptron* network (MLP), also synonomously labelled as a *feedforward neural network* (FFNN). An abstract view of an MLP is displayed in figure 2.2. The simplest form of MLP network consists of an *input layer* of perceptrons, also referred to as *neurons*, an *output layer* of perceptrons and an abitrary number of *hidden layers*. Each of the hidden layers described can contain any number of perceptrons, however the number of perceptrons contained within the input and output layers will typically match the number of input features and output classes respectively.

Given a vector of real-valued inputs $\mathbf{x} = [x_0, x_1, ..., x_N] \in \mathbb{R}^N$ and a vector of output values $\hat{\mathbf{y}} = [y_0, y_1, ..., y_D] \in \mathbb{R}^D$, an MLP network $\mathbf{f}$ is a mapping between two Euclidean spaces, i.e. $\mathbf{f} : \mathbb{R}^1 \rightarrow \mathbb{R}^L$. For a MLP network with an arbitrary number of hidden layers $[l_1, l_2, ..., l_L]$, this can be defined as a series of mappings connecting the input space to the output space, i.e.

$$f_1 : \mathbb{R}^{l_1} \rightarrow \mathbb{R}^{l_2} \ , \ f_2 : \mathbb{R}^{l_2} \rightarrow \mathbb{R}^{l_3} \ , ..., \ f_{l_{L-1}} : \mathbb{R}^{l_{L-1}} \rightarrow \mathbb{R}^{l_L} \tag{2.3}$$

For a hidden layer of k-neurons, the corresponding weight values are contained in a matrix $\mathbf{W}$ of size $k \times N$, and the bias terms as a vector $\mathbf{b} = [b_0, b_1, ..., b_k] \in \mathbb{R}^k$. For L hidden layers, the weight matrices and

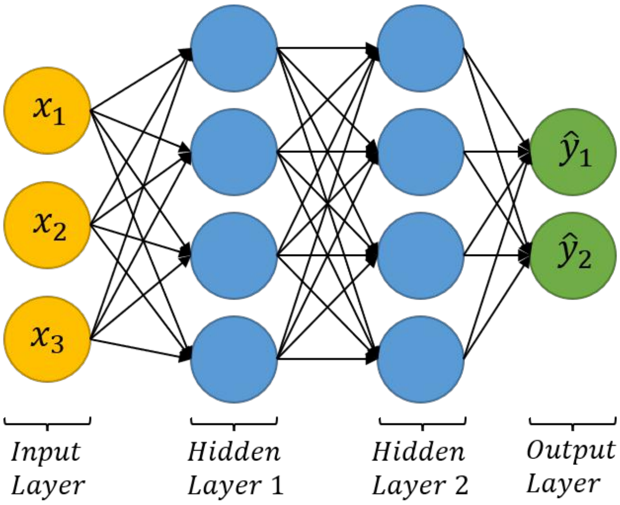


Figure 2.2: An abstract illustration of the layer-wise connections contained within an MLP.

bias terms are each contained as a set, where $\mathbf{W}_i = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_L\}$ and $\mathbf{b}_i = \{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_L\}$. Including an activation function σ, the mapping functions between layers is recursive, and can be defined as:

$$\mathbf{x} = f_i(\mathbf{x}) = \sigma(\mathbf{x}^T \mathbf{W}_i + \mathbf{b}_i) \tag{2.4}$$

Unlike the single perceptron, each neuron within the MLP network uses a non-linear activation function, to produce a real-valued output dependent to the input value. Sigmoid activation functions can be used here [13], which possess a typical 's' shaped curve and have a bell-shaped first derivative curve. However, there are many other activation functions which have uses, most notably rectified linear units (ReLU) or radial basis functions (RBFs). The logistic function is an example of a commonly used sigmoid activation function with upper and lower bounds of 1 and 0, i.e:

$$f(x) = \frac{1}{1+e^{-x}} \quad \rightarrow \quad \begin{cases} \lim_{x\to\infty} f(x) = 1 \\ \lim_{x\to-\infty} f(x) = 0 \end{cases} \tag{2.5}$$

Once each mapping has been completed between the input layer and hidden layers, the final mapping of $f_{L-1}(\mathbf{x})$ produces the vector containing the model output values $\hat{\mathbf{y}} \in \mathbb{R}^D$. For a regression task, the mean-squared error (MSE) loss function could be applied, given model output $\hat{\mathbf{y}}$ for each observed input and a set of correct output values $\mathbf{y}$. The loss value would subsequently be calculated as:

$$E(\mathbf{x}) \;=\; \frac{1}{D}\sum_{i=1}^{D}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 \tag{2.6}$$

For a classification task, the objective is to predict the probability of an input $\mathbf{x}$ with an associated label from a set of labels $\mathbf{y}_i \in \{0, 1, ..., C\}$. To create a class probability distribution Z from the model outputs $\hat{\mathbf{y}}$, a softmax function is applied element-wise to an output layer of C-neurons to ensure $\sum_{i=1}^{C} \hat{\mathbf{y}}_i = 1$, i.e.:

$$Z_i = Softmax(\hat{\mathbf{y}}_i) = \frac{e^{\hat{\mathbf{y}}_i}}{\sum_{j=1}^{C} e^{\hat{\mathbf{y}}_j}} \tag{2.7}$$

From here the log of Z can be taken with $\mathbf{y}$ being the target label for input x. This results in the *cross-entropy* loss value:

$$E(\mathbf{x}, \mathbf{y}) = -\frac{1}{D} \sum_{i=1}^{D} \log(Z_{i,\mathbf{y}_i}) \tag{2.8}$$

For the specific application of MLP networks towards classification of images, typically single channel greyscale images of an arbitrary bit-depth are used, however this is not exclusive and is for ease of demonstration. Individual pixel values can be taken in a sequential fashion from row-to-row, producing the input vector $\mathbf{x} = [x_1, x_2, ..., x_P] \in \mathbb{R}^P$, where P is the number of total pixels, consequently producing a P-dimensional Euclidean space for the input layer of the MLP network.

For smaller-sized images, this space could be very manageable, where MLP networks have been shown to be very effective at times [14–16]. However as image sizes approach the typical resolutions seen in modern-day image captures, the feasibility and complexity of the computed spaces increases dramatically to the point where for a reasonably-sized ML task, an MLP network may not be time-effective to optimize.

2.1.2 Convolutional Neural Networks

CNNs are a DL tool that has been popularized within the last decade or so, due to their capability to solve image-based ML tasks to a degree that previous ML models were unable to reach [17]. A CNN is made up of 4 main components. First is that of a *convolution* operation, whose purpose is to extract features using localized regions of pixels from an input image. Following this, a form of activation function is imposed to introduce non-linearity into the model. This is most commonly done in the form of using a ReLU operation, however any non-linear function such as a sigmoid or hyperbolic-tangent could be applied here instead. Thirdly, *pooling* operations are applied to reduce the subsequent dimensionality of each feature map created by the previous convolutional operations. These operations are defined as 'layers' in a CNN model, where convolution operations combined with their respective activation function form a *convolutional layer*, and are subsequently followed by a *pooling layer*. These are succesively repeated until finally, a *fully-connected* (FC) portion is applied at the tail-end of the model, which produces the model output and provides a classification result. A fully-connected portion tends to be an implementation of an arbitrary MLP network of some sort.

In more detail, the following describes a typical convolutional operation with associated activation function. For the sake of simplicity, we define a single-channel image X of size $W_0 \times H_0$ (the extension to multiple channels is fairly trivial), a non-linear activation function σ, and a square convolutional kernel K^l for layer $l \in \{1, 2, ..., L\}$ of size $k \times k$ which can contain set values for specific purposes (i.e. vertical edge detection) or can be initialised as arbitrary weight values to be optimized against a cost function. To perform the convolutional operation with applied non-linearity, an element-wise weighted sum operation is performed whose output is fed through σ, i.e:

$$x_{ij}^{l} = \sigma\left(\sum_{g=0}^{k-1}\sum_{h=0}^{k-1} K_{gh}^{l} x_{(i+g)(j+h)}^{l-1}\right) \quad , \quad \sigma(z) = \begin{cases} z : z \geq 0 \\ 0 : z < 0 \end{cases} \tag{2.9}$$

This is repeated until each area of $k \times k$ pixels has been passed by the kernel K^l, consequently known as a *filter pass* or *sliding window.* The resulting output of the convolutional operation x^l will therefore be of size $(W_0 - k + 1) \times (H_0 - k + 1)$.

The next layer to be applied in-sequence is the pooling layer, which takes a square region of $p \times p$ pixels and outputs a single value only. There are numerous ways in which this operation can be performed, most notably *max-pooling* and *average-pooling.* However, max-pooling is arguably the most common form of pooling used therefore will be utilised for this section. For max-pooling, the maximum value for each region of $p \times p$ pixels is taken, and the input matrix x^l will be reduced from a size of $W_1 \times H_1$ to $W_2 \times H_2 \equiv \frac{W_1}{p} \times \frac{H_1}{p}$.

Once all L convolutional and pooling layers have been applied in a series to the input x^0, the FC portion processes the extracted features and produces a classification result for the image. As previously stated, the FC portion is simply an MLP network as described in section 2.1.1, which is applied to the tail of the series of convolution and pooling layers. Here, the final input x^l is flattened to form a single column vector, prior to being processed through an MLP of a defined number of layers and neurons. Once processed, the vector $\hat{\mathbf{y}} \in \mathbb{R}$ of size C containing a single output value per possible class is produced. The index corresponding to the maximum value in $\hat{\mathbf{y}}$ is taken as the observed class for input x^0, and the model with kernel and FC weight parameters can therefore be optimized against a chosen cost function $E(x^0, \mathbf{y})$ with input and true target data.

2.1.3 Review Of CNN Architectures

In this section, a brief review of literature surrounding prominent CNN architectures will be conducted. Alongside this, a short discussion of the suitability of each method for the task application of real-time object detection for chapter 3 is provided. This is done to ensure our results are consistent with the intended task and not tailored towards a particular outcome in any way. This will also be clarified in further detail in chapter 3.

There exists image classification models that could be used for the following experiments, which are very capable of producing high accuracy values in their respective tasks. DL models such as GoogLeNet [18] and ResNet [19] have both won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2014 and 2015 respectively. These models performed extremely well for their time, with ResNet achieving an error rate of 3.6% in the challenge. Since then, further developments of image classification models have occured, namely through densely-connected blocks [20], as well as widely-connected residual inception modules [21]. Whilst the classification performance of such methods can be extremely high, they lack the capability for object detection on their own, so would not be a suitable fit for the role required in this research.

In alternative developments, efforts have been made to produce DL systems capable of extracting localized objects from images, rather than capturing the image in its' entirety. This type of classification, referred to as object detection, is much more applicable to the task of smart homecare as relevant objects within the home envrionment can be detected and measured in relation to the position and movement of the resident. Early object detection models of R-CNN [22] and its' extensions, Fast R-CNN [23] and Faster R-CNN [24] provide a good real-time capability of classification, whilst maintaining respectable classification accuracy values. The most recent of the three, Faster R-CNN, increased the real-time capability of the the series to processing 5 frames-per-second (FPS) for an image resolution of 224x224 pixels, whilst maintaining a mean average-precision (mAP) of 73.2. This was done through the implementation of a region-proposal layer, which gathered the final feature map produced and subsequently proposed regional areas for objects.

As a further extension to this, the authors proposed Mask R-CNN [25], which assumes the role of producing detections at a pixular level, rather than image-wide. The resulting area detections can then be semantically segmented and labelled as such to produce a region that is classified as a particular object. This type of object detection may have potential to be adapted for use in applications such as dense human pose estimation [26], that can be further modified towards a fall detection platform. However, the processing speed of this model is limited in comparison (5 FPS) and is not necessarily suitable for the objectives of this research.

To account for the shortfalls of many DL models discussed, regression-based models may provide a suitable middle ground relevant to the task application. An early regression-based model YOLO [27] provides the capability of bounding box predictions for the objects detected over a prediction confidence value, whilst doing so at a respectable rate of classification. In its' proposal, YOLO was capable of achieving an mAP of 63.4 on the Pascal VOC 2007 dataset [28], whilst processing at a rate of 45 FPS for input image resolutions of 448x448 pixels. Following the success of YOLO, YOLOv2 [29] was published shortly afterwards as an improved version, producing a highest mAP of 78.6 with a processing rate of 40FPS on the VOC 2007 dataset. To do this, YOLOv2 implemented alternative ideas such as batch normalization, adjusting training and detection image resolutions, and the introduction of anchor boxes.

2.2 Introduction to Quantum Computing

2.2.1 The Qubit & Quantum Data Representation

To introduce quantum computing, again it is best to start at the lowest form of data, the quantum bit, or *qubit.* Where classical data is represented using bits contained as a state of 0 or 1, a qubit fundamentally represents quantum data as a 2-dimensional complex vector. *Dirac* notation is used throughout to denote a vector in an arbitrary complex vector space, i.e. $|\psi\rangle \in \mathbb{C}$.

The state space of a qubit, or *Hilbert* space, is represented using a set of orthonomal basis states, known as the computational basis states (CBS) $|0\rangle$ or $|1\rangle$, where:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.10}$$

When we measure a qubit, it falls into one of the CBS. However, a unique property within quantum computation is that we are able to define a quantum state as a *superposition*, or linear combination of these CBS. This defines a quantum state of a qubit by combining the CBS with corresponding *amplitude coefficients.* As a quantum state is fundamentally a complex vector, these amplitude coefficients are complex values [30]. Within quantum computing, we are also bound by the *normalization constraint*, where the sum of squares for these amplitudes must equal 1. With this knowledge, we can ultimately define an arbitrary quantum state $|\psi\rangle$ as:

$$|\psi\rangle \;=\; \alpha\,|0\rangle + \beta\,|1\rangle \;=\; \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad \textit{where} \quad |\alpha|^2 + |\beta|^2 = 1 \quad : \quad \alpha, \beta \in \mathbb{C} \tag{2.11}$$

The principle of superposition allows us to explore the state space of a qubit using an endless number of configurable quantum states. This provides us with a very powerful tool for representing and processing data in comparison with classical bits. When it comes to measurement, a qubit will fall into state $|0\rangle$ with probability $|\alpha|^2$ and state $|1\rangle$ with probability $|\beta|^2$.

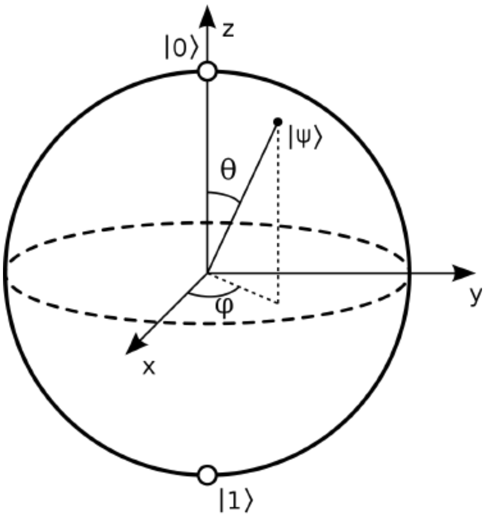


Figure 2.3: Bloch sphere representation of a qubit, as depicted in [30]

In order to comprehend how the state space of a qubit may function, we can use a visual model known as a Bloch sphere to represent how a qubit is geometrically defined. The Bloch sphere model can be seen in fig. 2.3. However, in order to match this representation, a quantum state must be redefined using different terms. Given a quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, since $\alpha, \beta \in \mathbb{C}$, they can be written in their exponential form $re^{i\phi}$, where r is the modulus of the complex number and (ϕ_0, ϕ_1) are parameters defining the extent of rotation. Thus, we can write $|\psi\rangle$ as:

$$|\psi\rangle = re^{i\phi_0}\,|0\rangle + re^{i\phi_1}\,|1\rangle \tag{2.12}$$

since

$$|e^{i\phi}|^2 = 1 \quad \rightarrow \quad |\psi\rangle = e^{i\phi}\,|\psi\rangle \tag{2.13}$$

$$\therefore \quad e^{-i\phi_0}\,|\psi\rangle = e^{-i\phi_0}(r_0 e^{i\phi_0}\,|0\rangle + r_1 e^{i\phi_1}\,|1\rangle) \tag{2.14}$$

Eq. 2.13 introduces the *global phase factor*, which is shown to have no effect on the observable quantum state. This can then simplify to:

$$|\psi\rangle = r_0\,|0\rangle + r_1 e^{i(\phi_1 - \phi_0)}\,|1\rangle \tag{2.15}$$

as $|r_0|^2 + |r_1|^2 = 1$. Using angle representations, we can deduce $r_0 = \cos\theta$ and $r_1 = \sin\theta$. Therefore:

$$|\psi\rangle = \cos\theta\,|0\rangle + e^{i(\phi_1 - \phi_0)}\sin\theta\,|1\rangle \tag{2.16}$$

In order to make the resulting quantum state match a complete spherical representation, it is convenient to define θ to $\theta/2$ so that $\theta \in [0, 2\pi]$. For tidyness, we can also define $\phi_1 - \phi_0 = \varphi$. Ultimately, this leaves us with the final quantum state representation:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \tag{2.17}$$

This state representation allows us to explicitly define a quantum state as a point anywhere on the Bloch sphere, and thus provides an extremely useful tool for visualizing the quantum state for a single qubit.

2.2.2 Quantum Information Using Multiple Qubits

As previously stated, we are able to represent a quantum state $|\psi\rangle$ as a superposition of the CBS $|0\rangle$ and $|1\rangle$, i.e, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. On its' own, a qubit is by definition a very powerful tool for computation. However, how does this extend to the use of multiple qubits? For a 2-qubit quantum state $|\psi\rangle$, we can express $|\psi\rangle$ as an expansion, or tensor product, of the CBS with corresponding amplitude coefficients for each individual qubit, i.e:

$$|\psi\rangle = (\alpha|0\rangle + \beta|1\rangle)(\gamma|0\rangle + \delta|1\rangle) \tag{2.18}$$

which expands to:

$$|\psi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \tag{2.19}$$

Identical to the quantum state of a single qubit, a quantum state for multiple qubits also follows the normalization constraint, where $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. During measurement, from eq. 2.19 it can be seen that the quantum state will fall into state $|00\rangle$ with probability $|\alpha\gamma|^2$, $|01\rangle$ with probability $|\alpha\delta|^2$, and so on. If we were to measure the probability that the second qubit would fall into state $|1\rangle$, regardless of the state of the first qubit, then this could be given as $|\alpha\delta|^2 + |\beta\delta|^2$.

As we extend our range of operations to multiple qubits, we want to incorporate another primary and unique aspect of quantum computing, *entanglement.* In short, entanglement allows for two or more qubits to have a state relationship that is depedent on the state that the other qubits are in. If we look at an arbitrary 2-qubit quantum state expansion as shown in eq. 2.19, then we have already seen an example of an entangled quantum state. If we let $\alpha, \beta, \gamma, \delta \neq 0$, then there is no way for that 2 qubit quantum state to be expressed independently of each others amplitude coefficients. Or in other words, we cannot express that 2 qubit quantum state as a tensor product of 2 individual 1 qubit quantum states.

From the previous 2-qubit expansion of the CBS, we can see that for an entangled quantum state, the amount of possible resulting states increases exponentially with the number of qubits used. In other terms, an n-qubit system can express 2^n ampltitude coefficients of information. This is where a large proportion of power is contained within quantum computing, and is a compelling reason why quantum computing should be explored.

Pure/Mixed States And Density Matrix Representation

When modelling quantum information and quantum computing, it is important to have an awareness of the concepts of *pure states* and *mixed states.* In the simplest terms, a system that is in a pure state is able to be precisely defined by a single state vector at any one time, whereas a system in a mixed state is not able to be defined by a single state vector, instead being an ensemble of different quantum states.

Throughout section 2.2, we have defined different quantum states in ket form as a superposition of CBS, such as $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ for example. If a qubit was initialized in an optimal environment to the state of $|\psi\rangle$, then the measurement outcome would be the state of $|0\rangle$ with probability $|\alpha|^2$, and state $|1\rangle$ with probability $|\beta|^2$. Even though this measurement process is probabilistic, we are certain that the qubit is in the state of $|\psi\rangle$ just prior to measurement, as there were no other influences that could possibly affect the state of the qubit during that time. Because the qubit is therefore represented unequivocally as $|\psi\rangle$, it can be defined as a pure state.

In mathematical terms, a pure n-qubit quantum state can be defined as:

$$|\psi\rangle = \sum_{i=0}^{N} \alpha_i\,|i\rangle \quad \textit{where} \quad N = 2^n \tag{2.20}$$

Unlike pure quantum states, mixed states are unable to be represented purely as a superposition of orthonormal basis state vectors. Instead, a mixed state is expressed as a statistical ensemble of multiple pure state outcomes, each with an associated probability. Again mathematically speaking, for a set length I of possible pure states $|\psi_i\rangle$ that individually have a p_i probability of occuring, a mixed state can be defined as:

$$\rho = \sum_{i}^{I} p_i\,|\psi_i\rangle\,\langle\psi_i| \tag{2.21}$$

This is known as a *density matrix* [30], and is a useful way of defining quantum states that can help track how a quantum system evolves over time when operations are performed. As an example of density matrix representation of a mixed quantum state, if we consider the following probability distribution containing two pure quantum states:

$$|\psi\rangle = \begin{cases} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & \text{with } p = 0.5 \\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) & \text{with } p = 0.5 \end{cases} \tag{2.22}$$

The density matrix equation formulated in eq. 2.21 can then be followed, giving:

$$\begin{aligned}
\rho &= \frac{1}{2}\left(\frac{1}{2}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\end{bmatrix}\right) + \frac{1}{2}\left(\frac{1}{2}\begin{bmatrix}1\\-1\end{bmatrix}\begin{bmatrix}1 & -1\end{bmatrix}\right)\\
&= \frac{1}{4}\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} + \frac{1}{4}\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\\
\rho &= \frac{1}{4}\begin{bmatrix}2 & 0\\0 & 2\end{bmatrix} = \begin{bmatrix}0.5 & 0\\0 & 0.5\end{bmatrix}
\end{aligned} \tag{2.23}$$

When using the density matrix representation for quantum states, it is useful to have an awareness of certain properties associated with density matrices. A particularly useful property is that of state purity, which can be obtained by taking the trace of the density matrix squared, i.e. $Tr(\rho^2)$. By doing so, a single value is obtained that determines the purity of the quantum state [30]. If this value is equal to 1, then the quantum state is pure, else if this value is less than 1, then it is a mixed state. Consider the following examples, using $|\psi\rangle$ from eq. 2.22 for example 2:

Example 1:

$$\begin{aligned}
|\phi\rangle = 0.8\,|0\rangle + 0.6\,|1\rangle &= \begin{bmatrix}0.8\\0.6\end{bmatrix} \quad , \quad \rho_\phi = |\phi\rangle\langle\phi|\\
\rho_\phi &= \begin{bmatrix}0.8\\0.6\end{bmatrix}\begin{bmatrix}0.8 & 0.6\end{bmatrix} = \begin{bmatrix}0.64 & 0.48\\0.48 & 0.36\end{bmatrix}\\
\rho_\phi^2 &= \begin{bmatrix}0.64 & 0.48\\0.48 & 0.36\end{bmatrix}^2 = \begin{bmatrix}0.64 & 0.48\\0.48 & 0.36\end{bmatrix}\\
Tr(\rho_\phi^2) &= 0.64 + 0.36 = 1 \quad \therefore \quad |\phi\rangle \text{ is a pure state.}
\end{aligned}$$

Example 2:

$$\begin{aligned}
\rho_\psi &= \begin{bmatrix}0.5 & 0\\0 & 0.5\end{bmatrix} \quad \rightarrow \quad \rho_\psi^2 = \begin{bmatrix}0.5 & 0\\0 & 0.5\end{bmatrix}^2 = \begin{bmatrix}0.25 & 0\\0 & 0.25\end{bmatrix}\\
Tr(\rho_\psi^2) &= 0.25 + 0.25 = 0.5 \quad \therefore \quad |\psi\rangle \text{ is a mixed state.}
\end{aligned} \tag{2.24}$$

2.2.3 Qubit Operations

Single Qubit Operations

Within classical computing, we operate on sets of bits using logic gates, i.e. a NOT gate, AND gate and so on. Quantum computation is no different in the sense that we use quantum gates to perform operations. These gates generally perform a set of complex reflections and rotations of the state vector around a respective axis. However, some of these gates are unique to quantum computation, and allow for highly complex operations to be performed.

For a single qubit, these quantum gates can be represented as a set of 2x2 unitary matrices. All quantum gate operations must be a unitary matrix, since unitary matrices preserve the length of the quantum state they are applied to, and we are bound by the normalization constraint described earlier. Three of the most basic quantum gates are based on the Pauli matrices, and are referred to as the X, Y and Z gate. These gates act on a single qubit only and perform rotations of π radians around their respective axes. The matrix corresponding to these operations are:

$$X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.25}$$

$$Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.26}$$

$$Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.27}$$

To represent how these gates affect a quantum state, we can demonstrate them using a Bloch sphere visualization. If we take a randomly selected quantum state $|\psi\rangle = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$ and apply the Pauli matrices individually, then by computing basic matrix operations the resulting states will be [30]:

$$X|\psi\rangle = \begin{bmatrix} 0.6 \\ 0.8 \end{bmatrix}, \; Y|\psi\rangle = \begin{bmatrix} -0.8i \\ 0.6i \end{bmatrix}, \; Z|\psi\rangle = \begin{bmatrix} 0.8 \\ -0.6 \end{bmatrix} \tag{2.28}$$

Plotting the resulting states using the Bloch sphere results in the following diagrams displayed in figure 2.4. Visualizing the operations performed acts as an extremely useful aid in comprehending what is occuring at a physical level. In this case, it is clear the operations that the Pauli matrices have had on the quantum state $|\psi\rangle$.

Another primary single-qubit gate is the *Hadamard* gate. This is a highly useful quantum gate which places a single qubit into an equal superposition of the CBS. The Hadamard gate can be represented as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.29}$$

The effect that the Hadmard gate has is that it will transform $|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. The resulting states is often be denoted by $|+\rangle$

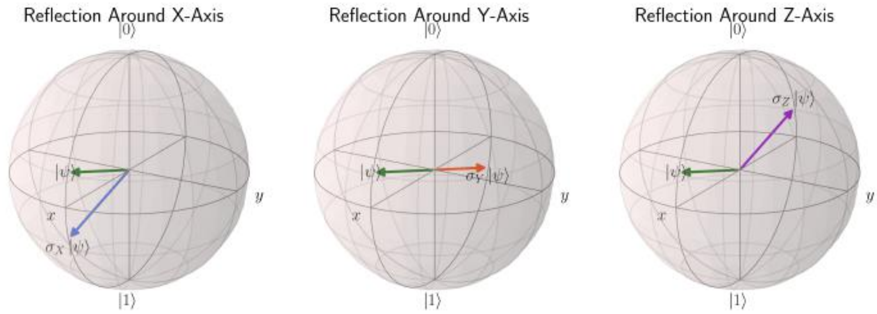


Figure 2.4: Bloch sphere visualizations displaying resulting transformations of the state vector $|\psi\rangle = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$ when Pauli operators are applied.

and $|-\rangle$ respectively, and are another set of basis states that are orthogonal in the x-basis, rather than $|0\rangle$ and $|1\rangle$ which are orthogonal in the z-basis. If we were to apply a Hadamard operation to a arbitrary quantum state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, then this would result in:

$$H|\psi\rangle \;=\; \alpha\,|+\rangle + \beta\,|-\rangle \;=\; \alpha\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{2.30}$$

So far, the single qubit operations that have been performed are static in nature, and do not allow for much flexibility in their range of rotation. When the Pauli matrices are deployed using their exponential form, we can define a set of rotation operators that rotate a state vector around their respective axis by a defined amount [30]. The corresponding rotation operators can be seen in equation 2.31.

$$\begin{aligned}
R_X(\theta) &\equiv e^{-i\sigma_X \frac{\theta}{2}} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right)\sigma_X = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\
R_Y(\theta) &\equiv e^{-i\sigma_Y \frac{\theta}{2}} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right)\sigma_Y = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \\
R_Z(\theta) &\equiv e^{-i\sigma_Z \frac{\theta}{2}} = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right)\sigma_Z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}
\end{aligned} \tag{2.31}$$

As before, it is useful to visualize the operations performed when applying these rotation operators. If we define a quantum state $|\psi\rangle = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$ and assign a random variable $\theta = \frac{5\pi}{4}$, the resulting rotations in their respective planes of movement can be seen in figure 2.5.

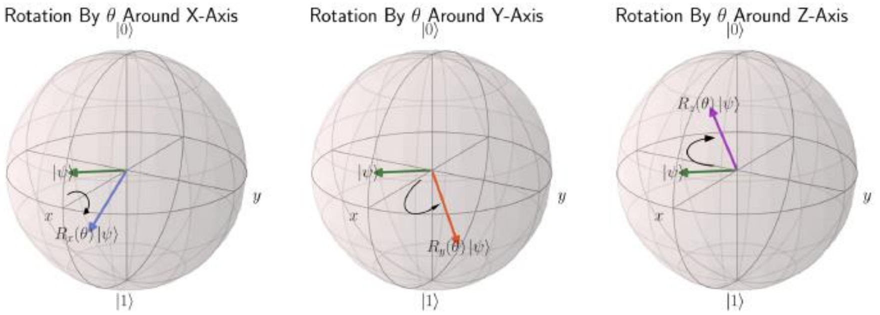

Rotation By θ Around X-Axis
$|0\rangle$
$|\psi\rangle$
x
y
$R_x(\theta)|\psi\rangle$
$|1\rangle$
Rotation By θ Around Y-Axis
$|0\rangle$
$|\psi\rangle$
x
y
$R_y(\theta)|\psi\rangle$
$|1\rangle$
Rotation By θ Around Z-Axis
$|0\rangle$
$R_z(\theta)|\psi\rangle$
$|\psi\rangle$
x
y
$|1\rangle$

Figure 2.5: Bloch sphere visualizations displaying resulting transformations of the state vector $|\psi\rangle = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$ when rotation operators are applied with a rotation extent of $\theta = \frac{5\pi}{4}$.

For a valid single-qubit gate operation, the only requirement is that it is a 2x2 unitary matrix. This means that an infinite set of possible gate operations can be defined, which are denoted as an abitrary unitary operation U. To realise an arbitrary unitary gate in practice, the *ZYZ-decomposition* reduces this unitary matrix into a combination of the previously defined Z and Y rotation matrices, with individual rotation parameters. For a set of parameters $\alpha, \beta, \gamma, \delta \in \Re$, an abitrary unitary U can be defined as:

$$U = \begin{bmatrix} e^{i(\alpha-\frac{\beta}{2}-\frac{\delta}{2})}\cos\left(\frac{\gamma}{2}\right) & -e^{i(\alpha-\frac{\beta}{2}+\frac{\delta}{2})}\sin\left(\frac{\gamma}{2}\right) \\ e^{i(\alpha+\frac{\beta}{2}-\frac{\delta}{2})}\sin\left(\frac{\gamma}{2}\right) & e^{i(\alpha+\frac{\beta}{2}+\frac{\delta}{2})}\cos\left(\frac{\gamma}{2}\right) \end{bmatrix} \tag{2.32}$$

Which, through typical matrix multiplication, can be decomposed to the following:

$$U = e^{i\alpha} \begin{bmatrix} e^{-i\frac{\beta}{2}} & 0 \\ 0 & e^{i\frac{\beta}{2}} \end{bmatrix} \begin{bmatrix} \cos\frac{\gamma}{2} & -\sin\frac{\gamma}{2} \\ \sin\frac{\gamma}{2} & \cos\frac{\gamma}{2} \end{bmatrix} \begin{bmatrix} e^{-i\frac{\delta}{2}} & 0 \\ 0 & e^{i\frac{\delta}{2}} \end{bmatrix} \tag{2.33}$$

Which is in the form of:

$$U = e^{i\alpha} R_Z(\beta) R_Y(\gamma) R_Z(\delta) \tag{2.34}$$

This is an extremely useful decomposition which allows for any possible single-qubit unitary operation to be decomposed at most into a set of three rotation gates at circuit level.

Multi Qubit Operations

In order to introduce entangled states between multiple qubits, we have to apply quantum gates that introduce the dependency needed between qubits. In the previous section, it was stated that a single-qubit quantum gate could be defined as a 2x2 unitary matrix. However, as we involve more qubits, these matrices also become scaled up in size. For a 2-qubit gate operation, we can define a arbitrary gate operation as a 4x4 unitary matrix. More generally, an n qubit operation can be defined as a 2^nx2^n unitary matrix.

One of the primary 2-qubit gates is the *Controlled-NOT*, or *CNOT* gate. The CNOT gate can be segmented into two portions, consisting of the *control* qubit and the *target* qubit. The CNOT gate conducts an X gate on the target qubit, depending on the state that the control qubit is in. If the control qubit is in state $|0\rangle$, then the CNOT gate will have no effect, however if the control qubit is in state $|1\rangle$, then the X gate operation will be applied. The matrix and corresponding effect on the quantum state vector $|\psi\rangle$ can be seen as.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}, \quad CNOT\,|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{bmatrix} \tag{2.35}$$

For a control qubit that is not in a superposition state, the CNOT has a trivial outcome. However, if the control qubit is in a superposition state, then the outcome of the CNOT gate can have some unique properties and results in a highly non-trivial state. As an example, consider a 2-qubit quantum state $|\psi\rangle = |00\rangle$. If we apply a Hadamard operation to the first qubit, then $|\psi\rangle$ is transformed to:

$$|\psi\rangle \;=\; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \;=\; \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \;=\; |{+}0\rangle \tag{2.36}$$

Individually, this state is nothing unusual, however, if we apply a CNOT gate with the first qubit being the control qubit and the second qubit being the target qubit, then this results in the following state:

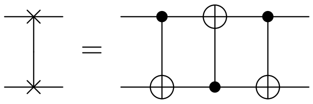
$$CNOT\,|\psi\rangle \;=\; CNOT\,|{+}0\rangle \;=\; \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.37}$$

The resulting state from this operation is known as a *Bell state*, and is an example of a maximally entangled state. If measured, this state has an equal probability of being measured in state $|00\rangle$ or state $|11\rangle$. Here, we can determine the output of both states from knowing the state that the other is in.

Occasionally, it is important to be able to exchange quantum states between two qubits within a circuit, and for this operation we can use another multi-qubit operation, the $SWAP$ gate.

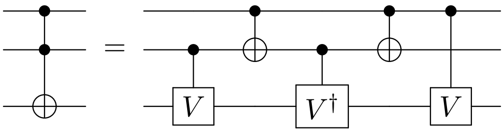
$$SWAP = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}, \quad |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}, \quad SWAP\,|\psi\rangle = \begin{bmatrix} \alpha \\ \gamma \\ \beta \\ \delta \end{bmatrix} \tag{2.38}$$

An interesting property of the aforementioned SWAP gate is that this can be broken down as a representation of three CNOT gates. I.e.



(2.39)

Within quantum computing, often it is required for 3 qubit operations to be performed. The first 3-qubit gate to be discussed here is the *Toffoli* gate. In essence, the Toffoli gate is a doubly-controlled NOT gate (which can also be referred to as a CCNOT gate), where 2 qubits are the control portions, with the third qubit being the target. The resulting action of this gate is that an X operation is performed to the target qubit, providing the 2 control qubits are in state $|1\rangle$. From a circuit perspective, the Toffoli gate (left-hand side) can be broken down using the Sleator-Weinfurter construction to a series of CNOTs and controlled-$\sqrt{X}$ gates:

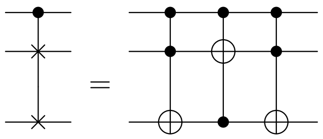


(2.40)

Where $V^2 = X$

Whilst this could be broken down further to single and multi-qubit gate operations, it is not necessarily relevant to this thesis and would mean explanation of additional quantum operations that do not appear in later sections. For now it is enough to be aware that a Toffoli gate performs a doubly-controlled NOT operation using 3-qubits.

Moving on, the second 3-qubit gate that will be introduced is the Fredkin gate, or controlled-SWAP gate (C-SWAP). Simply put, the Fredkin gate performs a SWAP operation on two qubits, providing the single control qubit is in state $|1\rangle$. Referring back to the 2-qubit SWAP decomposition to 3 CNOT gates displayed in eq. 2.39, the Fredkin gate can also be decomposed in the same manner, with additional control qubits added. This works out to be decomposed into three consecutive Toffoli operations, i.e:



(2.41)

2.2.4 Alternative Measurements

Qubit Measurement In Different Bases

Throughout section 2.2, measurement of a qubit has been conducted in the computational basis, where the state of each qubit will fall into state $|0\rangle$ or $|1\rangle$ dependent to the probability amplitude coefficient values. Generally, this is a popular scheme for qubit measurement as it results in a value of 0 or 1, which is convenient for many quantum computing uses whose nature revolves around binary-type measurements. This type of measurement is conducted with respect to the z-basis, or sometimes referred to with respect to the computational basis.

Due to the nature of quantum information representation, the z-basis is just one of many possible bases that a qubit can be measured against. Given that a set of desired basis states are orthonormal, then a quantum state can be measured with respect to the chosen basis. As an example, we can represent a quantum state $|\psi\rangle$ in the x-basis using the x-basis states of:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad , \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{2.42}$$

Where $|\psi\rangle = \alpha\,|+\rangle + \beta\,|-\rangle$. As before, the probability of measuring state $|+\rangle$ is $|\alpha|^2$ and likewise the probability of measuring $|-\rangle$ is $|\alpha|^2$. As a further example, the same principles apply when considering the orthonormal states in the y-basis, where $|\psi\rangle = \alpha\,|+i\rangle + \beta\,|-i\rangle$. i.e:

$$|+i\rangle = \frac{|0\rangle + i\,|1\rangle}{\sqrt{2}} \quad , \quad |-i\rangle = \frac{|0\rangle - i\,|1\rangle}{\sqrt{2}} \tag{2.43}$$

Projective Measurements And Expectation Values

Following on from measurements conducted in alternative bases, we can also take a *projective measurement*, or *expectation value* of a quantum state with respect to a valid quantum observable. This is a particularly useful type of measurement as we are able to gain flexibility in the possible quantum operations performed, and define a specific output state that we want to measure against. These can be thought of as the average possible outcome of all weighted probability measurements, however it is important to note that this value is not the result of highest probability that you would expect to see. Take a single-qubit quantum state $|\psi\rangle$ and a Hermitian operator O. The expectation value of this measurement is given by:

$$\langle O\rangle \equiv \langle\psi|O|\psi\rangle \equiv \sum_i a_i \langle\psi|P_i|\psi\rangle \equiv \sum_i a_i \langle\psi|i\rangle \langle i|\psi\rangle \tag{2.44}$$

Where $\sum_i a_i |i\rangle\langle i|$ is the spectral decomposition of O with eigenvalues a, i.e. $O = \sum_i a_i |i\rangle\langle i|$. The furthermost right-hand equation shows that the expectation value of $|\psi\rangle$ with respect to O is the sum of all possible measurement outcomes (the eigenvalues) weighted by the probability that this eigenvalue occurs if we are in the quantum state $|\psi\rangle$. As an example of how we take a projective measurement in practice, consider a single-qubit quantum state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the Hermitian operator Z, which equates to the Pauli-Z gate directly. If we take the spectral decomposition of Z, this results in:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \rightarrow \quad \begin{cases} |0\rangle\langle 0| = \text{eigenvector } |0\rangle \text{ with eigenvalue 1} \\ |1\rangle\langle 1| = \text{eigenvector } |1\rangle \text{ with eigenvalue -1} \end{cases} \tag{2.45}$$

Relaying this information into the projective measurement equation, the resulting expectation value is:

$$\begin{gathered} \langle\psi|Z|\psi\rangle = \langle\psi|0\rangle \langle 0|\psi\rangle - \langle\psi|1\rangle \langle 1|\psi\rangle \\ \langle\psi|0\rangle \langle 0|\psi\rangle = \frac{1}{2} \quad , \quad \langle\psi|1\rangle \langle 1|\psi\rangle = \frac{1}{2} \\ \therefore \langle\psi|Z|\psi\rangle = (1)\frac{1}{2} + (-1)\frac{1}{2} = 0 \end{gathered} \tag{2.46}$$

Fidelity Measurement Between Quantum States

Whilst the measure of *fidelity* is slightly different to previously discussed forms of qubit measurement, it is a very useful operation that allows for us to judge similarity between two quantum states. Analytically speaking, fidelity is a measure of overlap, or 'closeness' between two state vectors or density matrices through the use of an inner product. For a set of two pure quantum states, $|\psi\rangle$ and $|\phi\rangle$, the measure of fidelity $F(|\psi\rangle, |\phi\rangle)$ can be defined as:

$$F(|\psi\rangle, |\phi\rangle) \;=\; |\langle\psi|\phi\rangle|^2 \;=\; |\langle\phi|\psi\rangle|^2 \quad \text{(i.e. is symmetric)} \tag{2.47}$$

In simple terms, as the fidelity of two quantum states approaches 1, the closer they are and as their fidelity value approaches 0, the further away they are from one another. As an example, the states of $|0\rangle$ and $|1\rangle$ point in completely opposite directions to one another, therefore they will have a fidelity value of 0. If we measure the fidelity between states $|0\rangle$ and $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, which are orthogonal to one another, their fidelity will be $\frac{1}{2}$.

Whilst the above example holds true for pure quantum states, the measure of fidelity does change somewhat when handling mixed quantum states. Consider two mixed-state density matrices, ρ_1 and ρ_2, then the measure of fidelity can be defined as:

$$F(\rho_1, \rho_2) = \left(Tr\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right)^2 \tag{2.48}$$

However, if we are explicitly measuring the fidelity between a pure quantum state $|\psi\rangle$ and an arbitrary density matrix ρ (pure or mixed by nature), then the equation for fidelity can be reduced down to:

$$F(|\psi\rangle, \rho) = \sqrt{\langle\psi|\rho|\psi\rangle} \tag{2.49}$$

2.3 Quantum Computing for Machine Learning

2.3.1 Quantum Machine Learning Overview

In the past decade alone, ML has seen a vast array of successful developments, that have translated successfully across to industrial applications [31–33]. However, the rise of computational intelligence has not been without its disadvantages. One of the key challenges faced within ML is that of the growing reliance towards high-powered GPU units, that deliver vast amounts of processing capability. This is a challenge as modern developments to GPU capabilities are slowing down, reaching physical limitations. Therefore, alternative solutions to combat this reliance are needed.

QML is an area of research and development maturing steadily at this current time. Making use of quantum-mechanical principles, quantum computing is the logical choice for development in the pursuit of effective ML algorithms that remain efficient in nature. Typically, the broad field of QML can be split into 4 dinstinctive categories based on how quantum computing is applied to ML. This is dependent to the form of data considered, as well as the nature of the task itself. As Aimeur introduces in [34], these four categories are displayed in figure 2.6.

The category of CC is synonomous with classical ML, where classical data is fed through a typical ML algorithm (such as MLP, or SVM) and executed via a classical processor. Although completely classically-based, this is a category of quantum machine learning for classical ML algorithms that are quantum-inspired. The bottom left category of QC refers to how classical computing and classical ML algorithms may benefit quantum-based methods of computing, such as ML-based quantum error correction [35]. The third category, located in the bottom-right, is QQ, which translates

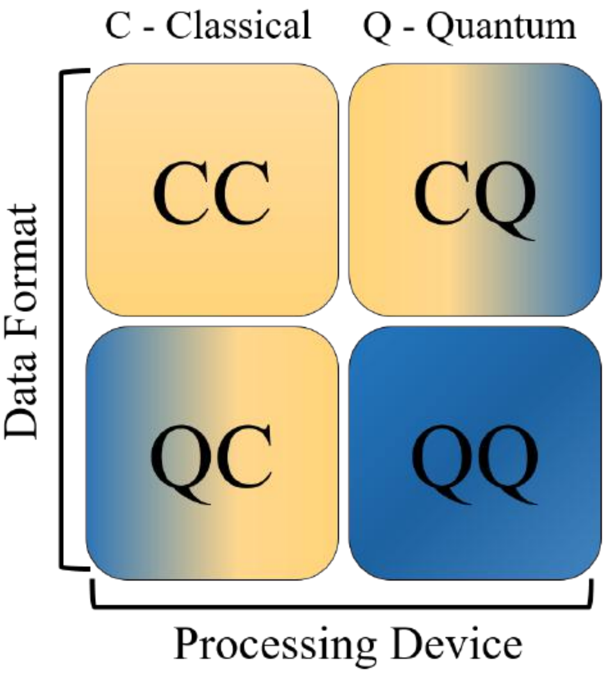


Figure 2.6: Categories of quantum machine learning dependent to the format of information and the processing unit implemented. CC - classical-classical, CQ - classical-quantum, QC - quantum-classical, QQ - quantum-quantum.

to quantum data processed via a quantum device. A typical process here may utilise the state of a quantum system as an input, prior to execution. This has been shown to provide benefits by removing the need for classical-based interventions which can be slow and costly [36], therefore improving efficiency when dealing with quantum data.

Finally, the category that lends itself to the focus of this thesis is CQ, the flow of classical data executed via quantum-based algorithms. This is arguably the area which has seen the most attention within QML so far [36], and traditionally follows a framework of some sort of encoding of classical information to its quantum form, followed by an execution of a designed quantum circuit routine. For supervised learning tasks, once measurement occurs the circuit output is fed back through a classical optimization loop, with the intent of minimizing an associated cost function of some sort. This makes it most similar to the flow of classical ML due to the availability of classical datasets, therefore providing an ease of transition for industrial applications. As image data itself is of classical format, logically this review of literature will focus within the realms of the CQ arena of QML.

There are several main challenges currently faced within the CQ area of QML. The first challenge is how we can effectively encode classical data into its quantum form, whilst ensuring the mapped feature space can maintain key information appropriately. The second challenge is how we should design quantum algorithms so we can perform useful transformations to encoded data. This is so we are able to extract key features that can be distinguished between via measurement. A third challenge within QML, although fairly generic, is where the main advantages lie by transitioning to quantum-based processing, and how substantial any of these advantages are.

As defined in [37], there are two approaches to QML development within the realm of CQ, the *translational approach* and the *exploratory approach*. The translational approach focuses towards the conversion of primitive classical ML methods into some form of quantum equivalent. This is done in the hope of gaining some form of advantage, be it through improved classification capability or complexity reduction. On the other hand, the exploratory approach attempts to investigate what particular ML tasks may be firmly suited to the physical characteristics of the quantum system used. As the realm of this thesis is directed towards the development of quantum-based image classification, it is therefore suitable to direct the attention of this review of literature towards the translational approach.

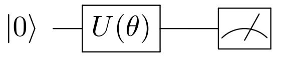
With this in mind, the objective of this review of literature is to provide a somewhat comprehensive overview of relevant QML methodologies that have been proposed, as well as to provide insight into the current schools of thought for the development of QML algorithms for classical data. Although there exists un-supervised methods targeted towards image classification [38–40], it is often considered a supervised learning task and will also be thought of as such within this thesis. For classical ML, the category of supervised learning can be seperated into two distinct sub-categories of algorithm, regression algorithms and classification algorithms. The aim of a regression algorithm is to establish a relationship between a set of independent variables to a dependent variable. In contrast, the aim of a classification algorithm is to assign a set of data with one of many defined category labels. Therefore, two sub-areas of review will promptly be conducted towards proposed quantum regression methods, as well as proposed quantum classification algorithms. However, firstly is it relevant to introduce a method of application for quantum computing to machine learning, the variational circuit model.

2.3.2 Variational Circuit Model

The following section will discuss a particular method of application for quantum machine learning, based on parameterized circuitry. This section will be used as a showcase for how a quantum circuit may be optimized, using the example of an arbitrary unitary gate U. Whilst in the cases shown, U takes the form of a z-rotational gate, the principles remain the same and extend to arbitrary quantum circuits as a whole. As seen previously in eq. 2.31, the matrix representation of an R_Z gate is written as the following.

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad , \quad R_z(\theta) = e^{-i\sigma_z \frac{\theta}{2}} = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{2.50}$$

The matrix on the left hand side represents a fixed rotation around the Z-axis of π radians. The matrix on the right hand side is the Euler decomposition of a rotation around the Z-axis, with a rotation extent of $\theta/2$ radians. By parametrizing the extent of rotation, we can then represent a quantum circuit as a function of its' parameters θ. Consider a single-qubit circuit, initialised in state $|0\rangle$. A z-rotation gate (denoted by $U(\theta_i)$ where $U(\theta_i) = R_z(\theta_i) = e^{-i\sigma_z \frac{\theta}{2}}$ is implemented within the circuit with parameter θ_0. This results in the following circuit, which is depicted in Dirac notation by an initial starting state $|0\rangle$ followed by the described unitary operation U of argument θ and a measurement operation.



(2.51)

When we measure the expectation value of $f(\theta)$ with respect to a hermitian observable $\hat{A}$, we can consider the circuit as the equation $f(\theta) = \langle 0|U^\dagger(\theta)\hat{A}U(\theta)|0\rangle$. We can then compute the derivative of $U(\theta)$, such that:

$$\frac{\partial}{\partial\theta}U(\theta) = -\frac{i}{2}U(\theta)\sigma_z \tag{2.52}$$

When $\frac{\partial}{\partial\theta}U(\theta)$ is substituted back into $f(\theta)$ and we follow the commutator rule $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$, we can derive $\nabla_\theta f(\theta)$ in terms of the circuit function f:

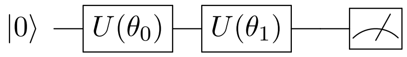
$$\nabla_\theta f(\theta) = \frac{1}{2}[f(\theta + \frac{\pi}{2}) - f(\theta - \frac{\pi}{2})] \tag{2.53}$$

This is known as the *parameter-shift rule* [41]. More generally, the parameter-shift rule can be defined as:

$$\nabla_\theta f(\theta) = r[f(\theta + \frac{\pi}{4r}) - f(\theta - \frac{\pi}{4r})] \tag{2.54}$$

Where although r has been used in an earlier definition, r is an arbitrary multiplier in this case. Therefore, in order to calculate the gradient of a circuit function f containing a single gate with respect to the input parameter, we require two circuit evaluations.

The above shows the case of evaluating the gradient for a single-gate quantum circuit. However, as we extend our circuit to multiple gates, the parameter-shift rule will remain constant throughout. If we consider a single-qubit quantum circuit initialised in state $|0\rangle$, and implement two z-rotation gates (denoted by $U(\theta_i)$) with parameters $(\theta_0, \theta_1) \in \mathbb{R}$. This results in the following circuit:



(2.55)

Using the parameter-shift rule, each unitary operation can be evaluated one at a time. By starting with:

$$f(\theta) = \langle 0|U^{\dagger}(\theta_0)U^{\dagger}(\theta_1)\hat{A}U(\theta_1)U(\theta_0)|0\rangle \tag{2.56}$$

in order to evaluate the $\frac{\partial}{\partial\theta_1}$, we can absorb θ_0 into the initial state of the circuit $|\psi\rangle$, which leaves us with:

$$\nabla_{\theta_1} f(\theta) = \langle\psi|\nabla_{\theta_1} U^{\dagger}(\theta_1)\hat{A}U(\theta_1)|\psi\rangle \tag{2.57}$$

The parameter-shift rule can then be applied to the resulting state, as with a single-gate circuit. Likewise, in order to evaluate $\frac{\partial}{\partial\theta_0}$, the observable $\hat{A}$ can absorb θ_1, leaving a resulting state of:

$$\nabla_{\theta_0} f(\theta_0) = \langle 0|\nabla_{\theta_0} U^{\dagger}(\theta_0)\hat{\mathcal{B}}U(\theta_0)|0\rangle \tag{2.58}$$

where $\hat{\mathcal{B}} = U^{\dagger}(\theta_1)\hat{A}U(\theta_1)$. Again, this can be evaluated using the parameter-shift rule. As a more general rule for multi-gate quantum circuits, we can define the gradient of each gate to be evaluated as:

$$\nabla_{\theta_i} f(\theta) = \langle\psi_{i-1}|\nabla_{\theta_i} U^{\dagger}(\theta_i)\hat{\mathcal{B}}_{i+1}U(\theta_i)|\psi_{i-1}\rangle \tag{2.59}$$

where $\hat{\mathcal{B}}_{i+1} = U^{\dagger}(\theta_N)...U^{\dagger}(\theta_{i+1})\hat{A}U(\theta_{i+1})...U(\theta_N)$, with N being the number of gates implemented within the circuit. In essence, to evaluate a gate at index i, any gate applied before i will be absorbed into the initial circuit state operation, and any gate will be absorbed into the measurement operator. By cycling through i, $\nabla_{\theta} f$ can be evaluated using successive parameter-shift operations, thus requiring two circuit executions per gate.

To recap section 2.3.2, by parametrizing the extent of rotations of operators, we are able to represent a quantum circuit as a function of its' parameters. By following the parameter-shift rule, gradients of parameters can be calculated dependent to the output measurement. With the inclusion of an objective function that the circuit outputs can be measured against, we can then optimize variational circuits to a specific task using *gradient descent*, or similar optimization schematics. This is an extremely important aspect, and forms the basis for many early developments of quantum machine learning algorithms.

2.3.3 Quantum-Based Regression

As a very brief recap of the premise of linear regression, for a set of data x (independent variables) and a second set of data y (dependent variable), the objective of linear regression is essentially to fit the data to the plane

$y = ax + b$. This is done with the intention of minimizing a loss function containing target outputs $\hat{y}$. The following least-squares method is a tried and tested method of doing so [42–45]:

$$E(x) = \sum \left(y - \hat{y}\right)^2 \tag{2.60}$$

For a linear regression problem, the cost of computation naturally rises with the rise of task dimensionality. However as linear regression boils down to a set of matrix operations, quantum computing may provide a logical choice for improving efficiency of computation as matrix operations are fundamental within quantum operations [30].

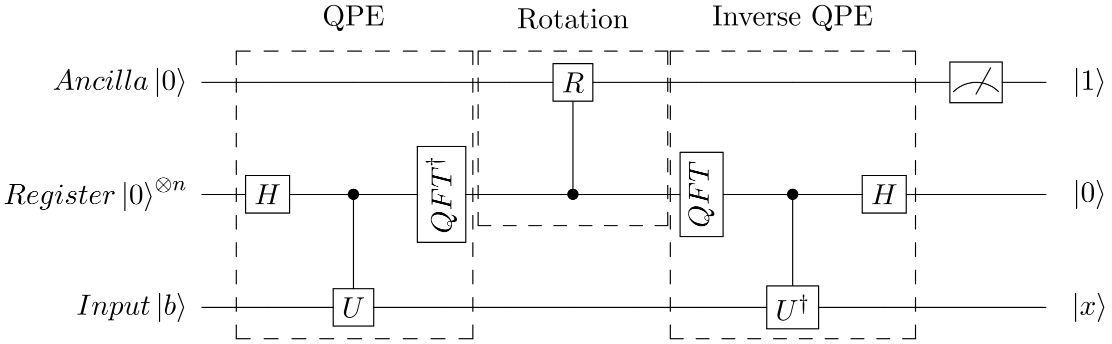


Figure 2.7: Circuit view for the HHL algorithm. Once an input state b is encoded, quantum phase estimation (QPE) occurs which includes a quantum fourier transform (QFT). This is followed by controlled rotation operations and an inverse QPE. Finally, measurement occurs and x can be decoded.

Harrow et al designed an early quantum algorithm in order to solve systems of linear equations, known as the HHL algorithm [46]. For a linear system of equations $A\vec{x} = \vec{b}$, the HHL algorithm encodes $\vec{b}$ within the amplitudes of an equivalent quantum state. Following this, quantum phase estimation occurs to estimate the eigenvalues of A, which are then used in a series of controlled-rotation operations. Finally, a reverse quantum phase estimation operation is performed prior to measurement, which can subsequently be used to calculate $\vec{x}$. With this as inspiration, an improved version of the HHL algorithm was presented as part of a solution to fit a least-quares regression model [47]. Here, the method consisted of different subroutines for preparing the quantum state, estimating the closeness of fit and the subsequent set of slope and bias parameters (a and b seen in eq. 2.60).

In other work, the technique of density matrix exponentiation as explored in [9] was utilised to determine the eigenvectors and eigenvalues of an arbitrary density matrix via quantum phase estimation. This was implemented in an alternative quantum linear regression algorithm [48], under the notion of a quantum perceptron unit.

Alternatively, Wang states that it may be suitable to consider cases where the number of data samples is very large in comparison to the number of independent variables [49]. With this in mind, rather than store the optimal fitting parameters as a quantum state, like in [47, 48], Wang output the fitting parameters classically for future usage on unseen data with little additional cost [49].

Again, extending from the HHL algorithm and that was proposed in [47], an alternative quantum least squares regression algorithm was proposed in that runs in $O(\log(n))$ time [50]. In this work a proposition for a quantum ridge regression method was also included. Furthering this was a quantum ridge regression algorithm proposed in [51] based on k-fold cross validation [52], a technique widely used across classical ML [53–55]. This work was the first to use quantum k-fold cross validation to determine a suitable regularization parameter. Alongside this, when contained as a quantum state, the estimated fitting parameters could predict new data exponentially faster than an equivalent classical method for low-rank design matrices containing the independent variables x.

As an additional note, it is interesting to point out that the translational approach to QML is not one-way. A train of thought has emerged within classical ML research towards development of a new innovative set of algorithms that are *quantum-inspired*. For example, although linear regression is a tried-and-tested method within classical ML, [56] takes inspiration from the HHL algorithm and applies it in a classical setting where stochastic gradient descent (SGD) can be applied for optimization. Like the translational approach for classical ML to QML, quantum-inspired algorithms attempt to convert techniques utilized within QML back across to improving classical methods. Whilst this review will not go into depth with this category of algorithms, it is just an interesting point to state that it may be beneficial to exchange information and methods between QML and classical ML, rather than segregate the two fields from one another.

2.3.4 Quantum-Based Classification

Within classical ML, a logical advancement from linear-regression algorithms is that of a support vector machine (SVM), therefore it makes sense to discuss proposed QML-based SVM algorithms. Given a set of n-dimensional data, where n represents the number of features, the primary objective of an SVM is to determine a n-dimensional hyperplane that maximally-seperates the set of data into their respective classes.

As an SVM is fundamentally a magnitude of linear algebraic oeprations, quantum computing may be a natural modelling solution due to the linear nature of quantum operations. An early Quantum SVM method proposed in 2014 demonstrated the potential for QML, by implementing their algorithm in $O(\log NM)$ time for both stages of optimization and classification [57]. To do this, they expressed the SVM as an approximate least-squares problem as in [44]. Their solution then utilised the density matrix exponentiation as proposed in [9], followed by the HHL algorithm to describe the subsequent hyperplane in a quantum state. The classification result is then obtained from a swap test between the state describing the hyperplane and the state containing input data.

In other areas, developments have also occured towards proposals of quantum k-nearest neighbour (k-NN) algorithms. To summarise a k-NN algorithm, a datapoint is classified subsequently as its nearest pre-determined k datapoints based on a metric of distance. Typically, classical k-NN algorithms use Euclidean distance as a primary measure [58–60], however many other distance measures can be utilised. As highlighted in [61], the calculation of overlaps for N-dimensions in a quantum manner is much more efficient in $O(\log N)$ time that the classical equivalent of $O(N)$ time.

A particular measure of overlap explained in section 2.2.4 is that of fidelity. Related to the cosine metric of distance [62], fidelity is a measure of closeness between two quantum states. To calculate fidelity, a swap test routine can be conducted. This is implemented within a quantum k-NN algorithm [63], where the fidelity of input states encoded via amplitude encoding was calculated to determine a classification result. Rather than calculate fidelity independently, an oracle was applied to the dataset stored in a superposition state to calculate fidelities simultaneously.

In another work [64], input data was encoded via amplitude encoding to form a superposition state. A swap test was then conducted to calculate the distance between the contained vectors. Following this, the authors searched for k-nearest distances via a method in [65] that utilises iterations of Grovers algorithm. This implementation of these quantum subroutines enabled a computational complexity of $O(\sqrt{kM})$ for k neighbour calculations and M datapoints over the classical equivalent of $O(M \log k)$.

Similarly to this, Adu-Gyamfi proposed a similar methodology where the dataset of images are first converted to a series of 0s and 1s via a threshold prior to being encoded into a superposition via basis encoding [66]. A swap test was then used as a measure of distance, followed by amplitude estimation and a Grovers search to determine the k most similar points. Their method improved runtime complexity over existing algorithms to $O(R\sqrt{kM})$, requiring R oracle operations for amplitude estimation, k neighbour values and dataset of size M.

As mentioned, other measures of distance may be utilised within a nearest neighbour based algorithm. Naturally, this can be challenging for QML algorithms as we are limited to what means of distance that can be computed in a purely quantum manner. Whilst fidelity via a swap test tends to be a common choice, [67] proposed the use of Hamming distance as a plausible alternative. Hamming distance is a metric of comparison for two binary strings of equal length, and equates to the number of bit positions that are different in a direct comparison of the strings. To do this, the authors basis-encoded feature vectors of the training set in a binary format onto a subsequent set of qubits. Following this, a testing data sample was encoded identically as before, where a unitary operation was applied that calculated Hamming distances and stored the result in an ancillary qubit. Once measured, the classification result is obtained. Through an intuitive calculation of Hamming distance via quantum subroutines, the algorithm runs in $O(n^3)$ time for n-dimensional feature vectors, whilst outperforming previous algorithms of [61, 68] on a transformed MNIST dataset.

Although the quantum-based methodologies discussed until now do not specifically utilize variational quantum circuits (VQCs) as their primary backbone, it is important to relay them to show that VQCs are not the only application of quantum computing to machine learning. As discussed, many proposed advantages of quantum-based algorithms occur via naturally faster vector operations, such as calculation of inner products or state overlaps. However due to their optimization capability, VQC models have provided the basis for a wide variety of QML algorithm propositions in more recent times.

Remaining with the content of quantum SVM algorithms, an alternative implementation of a quantum SVM in [11] proposed to utilize the natural high-dimensionality of Hilbert space to represent the feature space for a classification problem as a quantum state. In order to do this, layers of parametrized unitary operations and entangling gates inspired by [69, 70] were sequentially performed to an array of qubits that generates the class seperation hyperplane. As a secondary proposition, the authors also presented a combined method that used a quantum computer to estimate the kernel function followed by a classical SVM for classification. However, within their work the authors pointed out that a quantum advantage could only be gained for feature maps that are classically hard to estimate.

During a similar timescale, the concept of encoding classical data as a complex feature Hilbert space has been simultaneously explored by others. Schuld determined that various mappings of data between Euclidean and Hilbert feature spaces were analagous to linear, polynomial and cosine kernels [71]. A classically hard to compute kernel could then be created through efficient quantum estimation of inner products, for example via a swap test. This could then be fed through a classical SVM, or similar method. A secondary approach explored was through the representation of the training model as a quantum state. The circuit that produces the model state could then by optimized via a hybrid manner against a chosen cost function.

Furthering this, [72] splits the notion of VQC-based quantum classifiers into 4 distinct processes of feature mapping, unitary transformation, measurement and post-processing. In their methodology, input data was encoded into an equivalent feature Hilbert space via amplitude encoding.

Then, a low-depth VQC consisting of various unitary operations was implemented prior to measurement of a single qubit before post-processing using a threshold to determine a clasification result. The architecture proposed enabled a low number of trainable weight parameters, which remained polylogarithmic in proportion to the dimensionality of the input. Experimental results also showed that the quantum algorithm performed well on a set of datasets from the UCI repository [73]. This was in a comparison against multiple classical MLP networks and SVM classifiers.

Utilising fidelity as a cost function, [74] proposes a setup for VQC that is analagous to classical MLP networks. In this work, sequential unitary operations were linked to typical neurons contained within an MLP, with varied width and depth from the number of input and output qubits dependent on the encoded data format.

Progressing from classical MLP networks comes CNNs, a part of deep learning. These types of networks have shown promising results on a variety of tasks, including image classification [75–77]. However, a large disadvantage to CNNs is that they can often be extremely time consuming to optimise and execute [78–81], which makes quantum-based CNNs a favourable pursuit to try and reduce this disadvantage.

To date, there have been proposals that use a backbone of VQCs that transform input data in a manner similar to CNNs prior to measurement. [12] proposes a quantum CNN architecture that mimicks the traditional convolutional, pooling and fully-connected sequence via two-qubit unitary operations applied to the input state as a convolutional operation. This is followed by measurement of alternate qubits combined for pooling, which also introduces non-linearity by using the measurement value as the parameter for an additional applied rotation operation. A unitary operation covering the remaining qubits then equates to a fully-connected portion. A fraction of the final qubits are measured to determine a classification output, where unitary operation parameters are then optimized against a cost function.

In a separate work, [82] proposes a quantum CNN that computes the forward pass as a product between an input and convolutional kernel via quantum inner product estimation. Non-linear activation, which is not trivial for quantum computers to implement, is applied using a boolean circuit as a non-linear function and the inner product output. Rotational operations and amplitude amplification routines are then performed, to enable higher-valued pixels to have a higher probability of being measured. Measured pixel values are then stored in quantum random-access memory (QRAM). QRAM uses a number of qubits to store an appropriate quantum state, for recall in future operations. The authors conducted simulations using MNIST data, where the algorithm was able to optimize in a similar fashion to a classical CNN architecture. However, the application of QRAM is still a largely discussed topic due to open questions surrounding its practical and physical implementation [83–87].

As the scale and workability of QML algorithms in general is fairly small, a hybrid approach is often considered as an intermediary step on the path to purely quantum-based methodologies. Hybrid algorithms implement a classical and quantum component together, and aim to combine advantages from each component to form a stronger classifier. A hybrid CNN classifier was explored in [88], where the introduction of quantum-extracted features, dubbed *quanvolutional layers* was explored. In the experiments, a circuit consisting of 9 qubits with a selection of unitary operations was assigned as a typical convolutional filter, where a single pixel was basis-encoded onto one qubit at a time via a threshold. The measurement output was then produced as a scalar value, and fed into a condensed CNN architecture.

Experimental results on MNIST data suggested that the quantum-extracted feature maps may provide a slight improvement to classification accuracy over the traditional classical setup. However there was little to distinguish between the third method examined where a random non-linearity was applied prior to execution, therefore it was unclear on any quantum advantage present over classical non-linear transformations.

In a different work [36], the authors propose numerous applications of transfer learning situated within the CQ (classical-quantum), QC (quantum-classical) and QQ (quantum-quantum) categories of QML. Arguably the largest application of interest at this time is the CQ application. To do this, the feature extraction portion (all layers minus the fully-connected portion) of the DL classifier ResNet18 [19] was used to process classical images in their natural form. The resulting feature maps were then executed via a *dressed* VQC consiting of alternating Hadamard, single-axis rotational and CNOT entangling gates. Expectation values taken in the z-basis then provided real outputs that correspond to each possible class via a classical linear layer.

Experiments were conducted using the Hymenoptera subset of the ImageNet dataset [7], where accuracy values of 96.7% in a simulated environment and 95% and 80% in quantum processing unit (QPU) environment were achieved. A second experiment was also conducted on two seperate subsets of data within the CIFAR-10 dataset [89] by retraining the network to the two tasks in turn. Here, the network was able to produce relatively good classification results of 82.7% and 96.05% to the tasks. The results in this work do support that knowledge can be transferred concurrently to the quantum domain through the use of dressed quantum circuits. However, it is unclear to what extent the notably deep classical feature extraction and final linear layer had on classification performance.

2.4 Summary

To summarise the review of relevant literature discussed throughout this chapter, there have been many developments towards effective QML algorithms. Within this, there appears to be two different objective paths that research is conducted towards, efficiency-based and performance-based. Efficiency-based developments tend to focus on reducing computational complexity of the algorithms classical equivalent, or performing a classically-hard task in a manner that is simple for quantum computers. However, performance-based developments may focus on improved classification capability via an appropriate metric such as accuracy or loss.

Quantum-based algorithms that appear to have the largest computational speedup over classical methods often seem to utilise quantum subroutines that are proven to be efficient, whilst varying the format that they are deployed. A fidelity-based approach is often considered here, where the naturally-efficient quantum estimation of inner products between states is used via a swap test in practice. The remainder of the algorithms circuit operations are considered task-dependently as either an optimizable VQC, or search-based via a suitable oracle routine.

For performance-based QML algorithms, typical approaches appear to consider the subsequent feature mapping between Euclidean and Hilbert space as a key factor in determining the highest performance. Encoding strategies that are commonly used include basis encoding, amplitude encoding and product encoding, which each have different qubit requirements to encode information fully. Findings of the discussed works support that quantum algorithms are capable of producing sufficiently complex features maps via an appropriate encoding method. However, the nature of information encoding is dependent to the format of the input data, and an optimal encoding strategy is largely unknown.

For QML algorithms that are VQC based, the nature of information encoding can also have a direct impact towards the depth or scale of the VQC, and subsequently the number of parameters to be optimized. Therefore, it is important to investigate potentially lucrative methods of encoding classical information that can create a highly-complex feature space, whilst minimizing the demands of the circuit through quantum operations and number of qubits required.

Overall, the literature discussed within this chapter shows the general trends appearing within QML developments. In order to tailor the approaches considered towards the application domain of image classification, it is appropriate to begin by considering the two key factors summarised here. Subsequent work within this thesis should investigate the usage of efficient quantum subroutines, as well as analyse information encoding strategies. This should be done with the intent of improving the complexity of Hilbert space feature maps whilst minimising quantum resources required.

With this said, it remains relevant to also investigate classical approaches to reducing the disadvantage to large-scale parametrization. One such method of doing so is the formerly mentioned transfer learning, therefore, chapter 3 will first focus on this area of research. This order of presentation is chosen since the bulk of the research focuses on quantum-based methods, therefore it is logical to introduce this first prior to progressing to purely quantum-based methods.

Chapter 3

Transfer Learning For In-House Envirionmental Sentience Based Smart Homecare

3.1 Introduction

The following chapter investigates the application of transfer learning techniques with a deep CNN architecture for a smart homecare application. This is done in an attempt to answer the following research question:

Research Question 1: *To what extent can the application of transfer learning maintain classification performance of deep CNNs, whilst reducing computational overhead from the optimization of large parameter counts?*

More specifically, in this chapter, transfer learning is applied to the DL object recognition architecture YOLOv2, to develop a system specific to the detection of objects commonly found within a home environment. Experimental results obtained showed promising classification accuracy values that are vital for the application domain. The potential for real-time was also considered, where the system was able to process images of 416x416 pixels at a rate of 10.7 and 2.0 frames-per-second on a GPU and CPU respectively. Overall, the research within this chapter shows that whilst the application of transfer learning can be effective at producing a high classification performance with reduced optimization load. However, alternative algorithmic developments should be pursued to ensure the computational resources available are used as efficiently as possible.

As previously mentioned, this chapter aims to address the objective outlined in research question 1, through the application and analysis of transfer learning techniques to an active area of research and development. It is important to target an area of research that provides a fair analysis of performance, therefore the chosen domain used for this research is smart home-care since this relies on both a high classification accuracy and a fast computational speed to be effective. The resulting system relies on lower computational overheads in order to provide the capability of rapid re-training dependent to the environment, as well as maintaining high levels of classification performance. With this said, the rationale for development in this setting and the resulting system performance is explained in more detail as the flow of the chapter progresses.

With the ageing global population increasing at a rapid rate, the number of population aged over 60 is expected to double by 2050 [90]. In 2017, approximately 1 in 8 people were aged 60 or over [90]. By 2050, this proportion is expected to increase further to 1 in 5 [90]. As a result of the increasing disproportion between people aged under and over 60 years of age, care-home workers may not be able to effectively provide an adequate amount of support to all residents within those homes, as the demands of the work will become too great. In the US alone, the predicted costs of medical care are projected to rise from 3% to 5.5% by 2050 [91]. To counter this, a promising solution is needed to assist in the care of the elderly population to not only reduce costs, but increase care efficiency overall globally.

In a survey report conducted by UNISON in 2015 [92], 49% of respondents reported that they did not receive adequate time with each resident, 45% felt that there were not enough staff to deliver appropriate care, and 65% stated that there was care being left undone because of understaffing problems. The issues faced here need a solution to allow care workers the capability to provide effective support to each individual person as efficiently as possible.

Similarly, approximately 28-35% of people aged 65 and over are victims of a serious fall at least once per year [93]. The likelihood of falling also only increases as individuals age further. Within this, elderly people residing within care homes are also much more likely to fall than their community-dwelling peers. In addition, a particular study [94] also showed that 15% of people who had fallen remained on the ground for over an hour. In relation to this issue, serious injury is found to be correlated with the time that a person remains on the floor unable to get up and back to their feet [93].

With the disproportion between care workers and the number of elderly people requiring care widening as time progresses, assitance and urgent medical attention when required may not be provided in a suitable timeframe. With hospitalization costs from fall-related injuries expected to increase in the US to $240 billion by 2040 [95], the capability to provide efficient care-giving for elderly persons may help to not only reduce the likelihood of serious injury, but also reduce medical-related costs associated with them. In order to achieve this, a form of smart home surveillance [96] is desperately needed for our ageing society. Whilst it is recognised that the usage of a smart-home surveillance system can be considered controversial on an ethical perspective, it should be noted that for the purposes of this work the ethical arguments for smart-home surveillance are not being discussed. The setting of smart home-care is being used purely for its algorithmic demands of high classification accuracy and fast computational speed.

DL is a modern subfield of machine learning, which focuses primarily towards deep artificial neural networks and their counterparts [97–102]. Recent advancements in the classification power of CNN methodologies have been substantial, since their introduction in 1998 [4]. As a result of recent improvements in the field of DL and CNN capabilities, the careful use of these networks has a large potential with many industrial applications, including those within healthcare and smart-living settings. In recent works, CNN methodologies have modelled promising solutions for human behaviour recognition [103], detection of falls in humans [104, 105], as well as the maintenance of independent living for citizens of senior ages [106]. The applications described here are all extremely important, and provide foundational aspects for efficient caring of humans.

In this chapter, the aim is to propose the use of DL to model a foundation capable of being tailored to smart home-care. To do this, the experiments conducted focus on the detection of common salient objects [107, 108] that are found within a typical home environment. To stay within the realm of the relevant research objective, this research is to analyse the applicability of a DL method for the real-world use case of smart homecare via transfer learning, with classification performance and rate of processing as the metrics discussed. To summarise the key findings of this chapter, experimental results determine that transfer learning is an effective tool for reducing much of the computational load during optimization, whilst possessing the capability to maintain a high level of classification performance.

This chapter is subsequently structured as follows. Firstly, the problem domain will be introduced, alongside justification for the use of DL as a potential solution to the problems discussed. Then, a brief review of relevant literature will be presented that discusses key image classification and object recognition architecures from recent years. Following this, the methodology used will be explained, and the setup of the chapters experiments will be described. Afterwards, the experimental results will be displayed and broken down for analysis. Finally, a discussion of the obtained experimental results will be given, where any relevant conclusions and propositions for future developments will be drawn.

3.2 Proposed Methodology

Based on the brief review of relevant literature in section 2.1.3, the following system design and experimental setup makes use of the deep CNN object detection framework YOLOv2 [29]. To summarise the network schematic, a total of 24 convolutional layers and 5 max-pooling layers are utilised as feature extraction. Within convolutional layers, kernel sizes range between a size of 3x3, 1x1 (linear weighting of pixels) using stride values of 1, where max-pooling layers utilise a 2x2 kernel size with a stride value of 2. Activation functions also consist of a leaky ReLU function [27] and are employed post convolutional operations.

Layer Type	Filters	Filter Size	Stride
Convolutional	32	3x3	1
Max Pooling	1	2x2	2
Convolutional	64	3x3	1
Max Pooling	1	2x2	2
Convolutional	128	3x3	1
Convolutional	64	3x3	1
Convolutional	128	3x3	1
Max Pooling	1	2x2	2
Convolutional	256	3x3	1
Convolutional	128	3x3	1
Convolutional	256	3x3	1
Max Pooling	1	2x2	2
Convolutional	512	3x3	1
Convolutional	256	3x3	1
Convolutional	512	3x3	1
Convolutional	256	3x3	1
Convolutional	512	3x3	1
Max Pooling	1	2x2	2
Convolutional	1024	3x3	1
Convolutional	512	3x3	1
Convolutional	1024	3x3	1
Convolutional	512	3x3	1
Convolutional	1024	3x3	1
Convolutional	1024	3x3	1
Convolutional	1024	3x3	1
Convolutional	1024	3x3	1
Linear	1	1x65	-
Linear	1	1x8	-

Table 3.1: Layerwise descriptions of the YOLOv2 architecture used for experiments within this chapter.

Prior to the experiments contained in this chapter, the base version of the network was pre-trained to the ImageNet 1000 class dataset [7]. To setup the following experiments and conduct transfer learning, the weights contained within the initial 20 convolutional layers that make up the bulk of the feature extraction portion are left frozen, whilst the final 4 convolutional layers and the fully-connected portion, which consists of 2 linear layers containing 65 and 8 neurons respectively, are initialized with random weight values sampled from a Gaussian distribution of mean 0 and standard deviation 1. The process of freezing these layers via transfer learning means that a $20/26 = 76.9\%$ reduction in layers to be optimized is attained, which subsequently does not take into account unparameterized pooling layers. The technique of batch normalization is also used throughout optimization, in order to assist with convergence whilst reducing the requirement for many alternative regularization methods to be applied [109].

For the data contained within this experiment, the Pascal VOC dataset [28] made up the majority of the training and test data. In its' original form, this dataset consists of a total of 20 individual image classes, however this is edited to leave 7 classes that are consistent with the aims of the experiment. This results in a dataset of images containing a person, bottle, chair, dining-table, potted plant, sofa, tv-monitor. In addition to this, a bespoke subset of data containing images of lamps was included. This was formed via a web-scrape of online images, prior to manually filtering and labelling those that were deemed suitable for the experiment. The combined dataset was then split accordingly into subsequent training and testing subsets, where table 3.2 displays the number of object class instances present in each. It should be noted that these numbers contain the number of instances that the object is seen, and does not equal the number of images in total. Because of this, there becomes a natural bias towards certain object classes as they are more likely to be contained in different ratios towards one another. For example an image may contain 4-6 chairs but only contain a single dining table, or a single image may contain a gathering of multiple people, yet no other objects contained within this list.

Training And Testing Set Class Instances			
Object Class	Training Instance	Testing Instances	Total
Person	17,401	211	17,612
Chair	3056	180	3,236
Bottle	1561	120	1,681
Dining Table	800	140	940
TV Monitor	893	127	1,020
Sofa	841	130	971
Potted Plant	1202	121	1,323
Lamp	486	73	559

Table 3.2: The number of total instances that each class of object appeared, and the number that appeared in the training and testing subsets of data. Note that this does not equal the number of images.

3.3 Results

In regards to optimization of the system, this was conducted at a timed basis of as many epochs possible within 40 hours. This equated to approximately 37 full epochs of the training data. For the selection of hyperparameter values, SGD [110] was implemented using a momentum value of 0.9, an initial learning rate of 1x10^{-5} that decayed by a factor of 5x10^{-4}. In order to track convergence and classification performance, 4 checkpoints were taken at random during training which coincided with training iterations 29,000, 69,000, 100,000 and 121,200. For each of these checkpoints, various metrics of classification performance were measured, where the results can be seen in table 3.3.

Iteration No.	Accuracy	Sensitivity	Specificity	Precision
29,000	59.52%	0.529	0.962	0.672
69,000	91.47%	0.910	0.997	0.975
100,000	87.93%	0.869	0.995	0.945
121,200	88.75%	0.884	0.997	0.976

Table 3.3: The overall classification performance metrics for each of the 4 model checkpoints saved.

From these results, it is clear that the model checkpoint of 69,000 training iterations produced the highest overall prediction accuracy value. Whilst these results support that the system was still converging to a satisfactory local minima during iteration 29,000, classification performance appears to drop for checkpoint 3 prior to a slight improvement once more for the final checkpoint. This behaviour may suggest that perhaps a slight overfitting occured during optimization, causing the subsequent test-set accuracy values to fall slightly. Regardless, The classification performance produced is still very strong considering the complex nature of the task.

	Object Class								
Metric	Person	Chair	Bottle	Table	TV	Sofa	Plant	Lamp	Average
True Positives	211	167	115	110	112	119	11	63	126
False Negatives	1	22	6	20	15	15	8	7	12
False Positives	0	7	0	10	0	2	2	3	3
True Negatives	820	842	916	912	920	905	919	966	900
Sensitivity (%)	99.5	88.4	95	84.6	88.2	88.8	93.3	90	91
Specificity (%)	100	99.2	100	98.9	100	99.8	99.8	99.7	99.7
Precision (%)	100	96	100	91.7	100	98.3	98.2	95.5	97.5
Accuracy (%)	99.9	97.2	99.4	97.1	98.6	98.4	99	99	98.6

Table 3.4: An individual breakdown of classification performance per class for the model checkpoint as 69,000 training iterations.

Table 3.3 displays a class-based overview of classification performance for the highest performing model. A confusion matrix for each individual class was determined from the testing subset of data. From here, various metrics of classification performance were derived, using the typical formulated approach based on the amount of true or false positive and negative object predictions. For clarity, the equations relating to these metrics are as follows:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \qquad \text{Specificity} = \frac{TN}{TN+FP}$$

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Where $TP =$True Positives, $TN =$True Negatives, $FP =$False Positives and $FN =$False Negatives. In relation to the results displayed in table 3.3, it is evident that the best performing class is the detection of people. This may be due to the number of training instances in relation to other classes, however only a single mis-prediction is a very promising result. The lowest performing classes are that of table and chair, with accuracy values of 97.1 and 97.2 respectively. These classes also held the highest number of false negatives and positives between them in comparison to others. Arguably, the detection of these objects may be more difficult than others, as they are often partially hidden in view, or are obstructed by other objects around them. This may make these objects harder to distinguish than others which may be more likely to be in full view such as the bottle class for example, which would likely be directly placed on top of a table etc.

Interestingly, the classification performance for the lamp class is within the top 50% of classes for accuracy. As the lamp data was much less substantial than other classes, this could be expected to perform to the lower end.

This supports the idea that transfer learning can improve performance on unseen classes of data more than if this system was trained completely from scratch, since the lamp class possesses a lower amount of false predictions than others. However, in contrast, the number of object instances in the testing subset was lower for the lamp than others, so this may potentially create a slight bias in favour of performance for this class Regardless, the classification performance for each class is still very strong, with high values of specificity and precision.

The final performance metric that was tested for this system was that of processing capability once trained. For the entirety of this research, optimization was conducted using an MSI GTX 1060 GPU unit, with 4GB of memory allocated to the working system. Once training had been conducted, the system processing speed via the same GPU unit was measured using the standard metric of FPS. However, realising that the final application may not have the capability of a GPU, the speed of processing was also measured using an Intel I5 8600k CPU in its' factory setting. Table 3.3 displays the system speed with the processing unit used.

Processing Unit	System Speed (FPS)
CPU	2.0
GPU	10.7

Table 3.5: Rate of image processing values dependent on hardware used. The GPU used is an MSI GTX 1060 with 4GB of RAM allocated to processing. The CPU used is an Intel I5 8600k used in its' factory setting.

As a result of the processing speed tests conducted, as expected the FPS rate via a GPU unit is much higher than that of a CPU(approximately 5x higher). Whilst the processing speed via the GPU may still not be considered real-time, it is still high enough to be very capable within the application of smart homecare. However, the processing speed of the CPU is just too low to have the substantial capability for further adaptations within the final application.

3.4 Discussion

Within this chapter, transfer learning techniques were implemented using the deep, CNN-based object detection network YOLOv2. The pre-trained system was retrained to a new task with increased specificity towards detection of indoor objects commonly found within the home envrionment. From the experimental results outlined throughout section 3.3, the second model checkpoint taken at 69,000 training iterations showed to perform the highest overall at an accuracy value of 91.47%. Based on the class breakdown for the 69,000 model, the high sensitivity values would benefit the system greatly when applied within a home environment, as objects become naturally obstructed by one another such as example detections seen in figure 3.1.

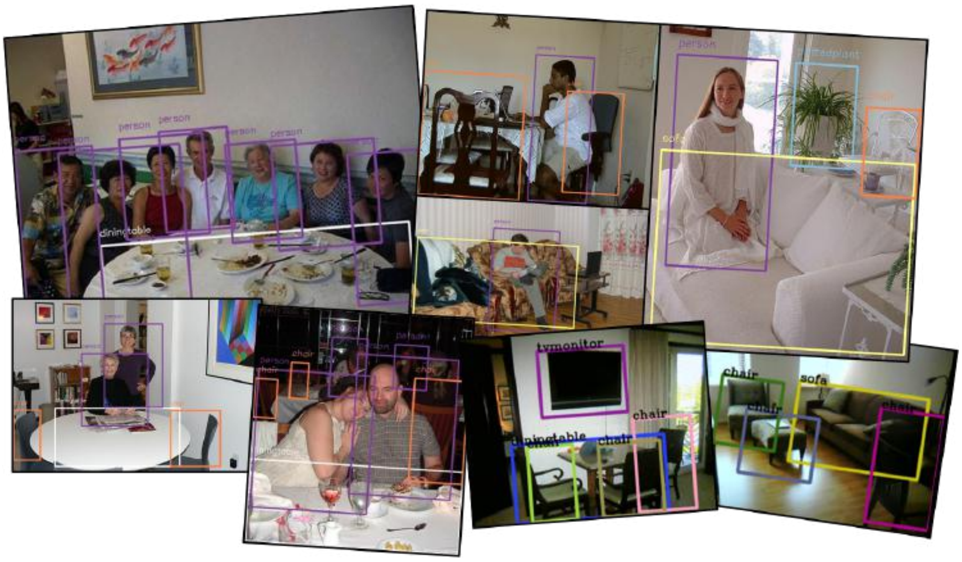


Figure 3.1: Multiple processed testing-set images that include various objects and the identified bounding boxes for detected objects.

In its' current state, the proposed system shows a great deal of potential to have applications within home care. The accuracy of detection is reliable enough to have different applications within smart assisted living. With further development, locations of detected objects may be able to be processed relative to one another, where a status could be enabled. For example, if a person was detected within a certain boundary of another object such as a chair, then the system status may change to 'sitting down'. This would naturally extend to a variety of other scenarios, such as drinking, watching tv or general movement. Sub-categories of movement such as falling over may also be suggested with further development, where alerts could then be triggered on a control device. In general, CNNs have been shown to generalize well to these categories of problems [104, 105], therefore showing promise for future development.

In relation to the growing demands of health and social care for the elderly population, additional features within the implemented system could allow for staff to allocate their time to residents more effectively than before. Residents that are of a higher priority due to medical and other reasons can then get the quality care and attention they require. This aspect of efficient caring is a large factor in not only improving quality of life for others, but also reducing the cost of care.

On discussion of the computationally expensive training of entire deep CNNs with many parameters, this was mostly negated through the use of transfer learning to just optimization of a handful of layers. The effectiveness of transfer learning is questionable as to whether the resulting classification capability will be suitable for its' end application or not. In the case of this research, the implementation of transfer learning was successful. The re-initialized end layers of the system were able to converge to a satisfactory local minimum on previously unseen object data. In doing so, individual class performances were relatively similar, even with uneven class representations in training data. Within just 40 hours of optimization, approximately 37 epochs of the training data could be conducted. With this timeframe in mind, the resulting classification performance from the application of transfer learning was very promising and the experiments objective was successfully achieved.

However, arguably not all of the performance metrics explored could be deemed suitable for task. Whilst the rate of processing measured on a GPU unit was fair, the rate measured via a CPU would simply not be fast enough for further application in a home environment. Whilst the hardware used to gather these results most certainly had a large impact, the hardware itself is not of a low quality by any means. On an industrial scale, the chances are the hardware will not be state-of-the-art in terms of processing speed, therefore this should be accounted for when evaluating proposed systems.

To summarise, the application of transfer learning enabled the algorithm to successfully retrain to the proposed task environment, producing high classification performance levels whilst reducing the need for optimization of the full CNN network. However the network implemented is still of a large scale, which is the common trend within the field of image classification and object recognition in recent years. Because of this, alternative developments towards image classification algorithms should be explored to ensure that we can produce the highest-performing efficient system overall. As DL systems have overtaken other algorithm developments within classical computing for image classification, quantum computing remains the natural logical choice for further pursuit. Investigation of the use of quantum computing for image classification will not only allow for well-informed insight into the potential computing power available, but the feasibility for such application on an industrial level.

3.5 Conclusion

With the growing shortage of quality care available for the ageing global population, ML and in particular DL has shown promising benefits that could become a potential solution to not only improve the quality of care given, but reduce costs simultaneously. However, due to the rising complexity of deep CNN algorithm structures it is not uncommon for the scale of optimization to update weight values in the millions. Transfer learning is a recently explored technique that aims to improve learning capacity and rate of learning on an unseen task by retraining the latter part of the algorithm only, rather than its' entirety.

In this chapter, transfer learning was conducted using the deep CNN YOLOv2 in conjunction with a bespoke dataset consisting of a subset of Pascal VOC data and custom-scraped images. This was done to explore the applicability of DL for a suitable use case of improving smart homecare, as well as investigate the effectiveness of transfer learning itself. Within the experiments conducted, the resulting classification accuracy of the developed system was promising, and is in a prime position for future developments to occur. However, the largest disadvantage for the system was the rate of processing itself, with a lowest value of 2 frames-per-second using a CPU.

Overall, the experiments conducted within this chapter show that transfer learning can be a very effective tool for the efficient retraining of deep CNNs. The need to optimize a full DL architecture from scratch was negated, leaving only a fraction of the originally large parameter count to be updated during backpropagation.The resulting classification performance on the retrained task was also seen to be very promising, with high levels of accuracy and sensitivity. The experimental results explored support the basic idea for transfer learning, in that classification performance can be maintained or even improved inter-task through the use of pre-trained feature extractors.

However, whilst the use of transfer learning may be effective, the field of image classification and object recognition is still stagnating with very large CNN architectures. Investigation towards alternative algorithm developments should be pursued to ensure that we are utlising the computing resources available as effectively as possible. Here, whilst the field fo quantum computing and quantum machine learning is in its' infancy, the natural efficiency of quantum information through quantum-mechanical properties may pave the way for a well-rounded image classification methodology.

Chapter 4

Non Gradient-Based Quantum Machine Learning For Facial Identification

4.1 Introduction

The following chapter presents research conducted to the development of a quantum computing algorithm targeted towards the application of facial identification in an effort to answer the following research question:

Research Question 2: *To what extent can a non-gradient based approach be taken to perform effective quantum image classification?*

To summarise this chapter, a quantum-based algorithm is explored that uses the quantum subroutine of a SWAP test. This is done to estimate fidelities of encoded feature vectors corresponding to images in order to determine a classification result via a threshold function. Here, computational complexity improvements of an exponential scale are able to be achieved in comparison to a classical SVM and k-NN, whilst sacrificing a small amount of classification accuracy in exchange. This is done through the efficient manner of which quantum information is encoded, in comparison to the computationally expensive methods of SVM and k-NN. Limitations of the method are also proposed, and areas for future investigation and development are considered to potentially improve the capability of the algorithm.

In recent years, QC has become an active area of research with the promise of many rich, performance enhancing benefits due to natural quantum-mechanical principles and behaviours. One of the most encouraging areas for development is QML. With the rise of QML being propelled towards desired benefits of performance speedups and rich feature spaces, certain developed quantum algorithms have proposed speedups of exponential factors over many classical algorithms currently used [9].

Facial identification is a highly critical task within ML that has many useful real-world applications. For humans facial identification is a simple task, however from a computer vision perspective the task is highly complex. Nevertheless, through the development and application of ML techniques, facial identification has been successfully applied to a wide variety of target domains that assist us in day to day living. Examples of these applications are plentiful across the modern world, such as biometric analysis [111], healthcare [112], personal security [113] and marketing [114]. Because of the high variety in applications, it is critical that we are able to innovate our approach to tasks such as facial identification in order to overcome drawbacks and challenges currently faced.

Currently, much of the focus within the noisy, intermediate-scale quantum (NISQ) era of QC and QML has been directed towards classification using variational circuit methods [12, 36, 69, 88, 115–117]. This is done as a step towards creating robust quantum neural networks [118, 119] and advanced quantum deep-learning algorithms [74, 120]. Whilst results of these developments have been promising, often a costly optimization step is required, which involves repeated executions of the quantum circuit to perform effectively. It is important to consider alternative methods of working to avoid a similar fate than that of classical deep learning, where the stacking of many layers is often seemingly done with the blind hope of improved classification performance, but at the expense of greatly increasing the computational effort required. Efforts have also been placed towards the development of quantum k-NN algorithms, that can conduct efficient searches of data stored in superpositions using oracle subroutines [67, 121]. However, there have not been many works that showcase quantum algorithms applied in practice to real-world tasks [122–124]. Many of the algorithms proposed rely on synthetic datasets for analysis [125, 126], or a purely theoretical works [57, 127–130] that will be difficult to validate until improvements can be made to the quantum hardware available.

In this chapter, the aim is to explore how we can use quantum machine learning techniques to build classifiers based on real-world applications, specifically towards facial identification in this case. Here, a classifier is built using foundational quantum modules that estimate fidelity between two individual quantum states. This is done using methods that are not variational-circuit based, to ensure the lengthy process of optimization can be negated. By doing so, an initial insight can be gained into how quantum algorithms may perform at a simple level, as well as gather perspective on where efforts may be placed to improve upon the initial results obtained to develop a robust classification system that has potential for real-world application. To summarise the research contained in this chapter, experimental results show that a fidelity-based approach is capable of effective classification, whilst removing the need for a resource-heavy optimization process.

The content in this chapter is therefore structured as follows. Firstly, the domain of the task will be introduced alongside justification for why a non-gradient based scheme of optimization is followed. Then, a brief review of common quantum information encodings is discussed. Following this, the methodology of the system is explained from start to finish. Following this, the experimental results will be outlined and an analysis will be provided in relative comparison to commonly used classical ML algorithms, in particular a support vector machine (SVM) and a k-NN. Finally, conclusions are drawn and any future directions for investigation and development are considered.

4.2 Methodology

4.2.1 Qubit Encodings

Review Of Quantum Feature Spaces

In classical computing, the fundemental unit of information is the bit. However, in quantum computing, this unit is known as the qubit. In their most basic forms and in terms of a single qubit, information is represented in a two-dimensional complex vector space, known as Hilbert space. When compared, whilst a set of n-bits can represent a single state (e.g. integer up to 2^n), a maximally-entangled set of n-qubits can represent 2^n states concurrently. This capability allows us to represent classical units of information in a complex, yet very efficient manner. By adapting our encoding of quantum systems, we can create feature spaces that are not tractable or feasible to replicate in classical systems.

In order to create these feature spaces, classical information must first be transformed into its' quantum equivalent form. Whilst a variety of quantum representations of classical data and images have been presented [131–134], two fundamental encoding methodologies are that of *Basis Encoding* and *Amplitude Encoding.* These schemes often form a basis for many other extensions and niche use cases, therefore it is only logical to define and explore these further.

Basis Encoding

Basis encoding can be seen as most similar to a binary-style representation of information in itself. Within basis encoding, the computational basis states of $|0\rangle$ and $|1\rangle$ are used to form a bit string that matches the classical input information. As an example, the 3-bit binary string $x = 010$ can be represented by the quantum state $|\psi\rangle = |x\rangle = |010\rangle$. More formally, consider an N-bit binary string $x_N = [y_1, y_2, ..., y_N]$, where $y_i \in \{0, 1\}$.

To consider this scheme in context, say we have N-datapoints $a_0, a_1, ..., a_N$, and F-features, $f_0, f_1, ...f_F$ that are binary in nature (e.g. yes/no, or $f_i \in$

$\{0,1\}$). In this example dataset containing 2 datapoints of 2 features each, $a_0 = [00]$ and $a_1 = [10]$. From here, we can represent the quantum state for each datapoint as a superposition of the computational basis states, giving the following quantum state $|\mathcal{A}\rangle$, i.e.:

$$|\mathcal{A}\rangle \;=\; \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \;=\; \frac{1}{\sqrt{N}}\sum_{n=0}^{N}|a_n\rangle \tag{4.1}$$

Amplitude Encoding

A different approach to the encoding of classical information to its' quantum equivalent is through amplitude encoding. Unlike basis encoding, here a normalized input vector is encoded using the associated amplitudes of a quantum state. As a simple example, say we have a two-dimensional datapoint x, with two features $f_i \in \mathbb{R}$, giving $x = [f_0, f_1] = [0.6, 0.8]$. Handily, this is already normalized since $|x| = \sqrt{0.6^2 + 0.8^2} = 1$. Therefore, this datapoint can be directly translated to its' equivalent quantum state, where each dimension is associated with an equivalent amplitude, giving the quantum state $|\psi\rangle = 0.6\,|0\rangle + 0.8\,|1\rangle$.

As a more generalised example, using amplitude encoding we can embed a single $N = 2^n$-dimensional datapoint using n-qubits by associating each dimension with their respective amplitude of the quantum state, i.e.

$$|\mathcal{A}\rangle = \sum_{i=0}^{N} f_i\,|i\rangle \tag{4.2}$$

In order to extend this to a dataset containing D data-points, then we simply extend the amplitude encoding scheme multiplicatively by DxN data-point dimensionality. For a single data-point of N-dimensions, the amplitude encoding scheme requires $n = \log_2(N)$ qubits. For a dataset containing D datapoints, this scheme extends to requiring $n = \log_2(D \times N)$ qubits to fully encode the dataset at one time. In summary, whilst basis encoding does have applications, amplitude encoding presents a much more efficient way of encoding classical information in comparison. Therefore, the amplitude encoding scheme will be used throughout this experiment.

4.2.2 Experimental Setup

Experiment Overview

The overall design of the system used for the following experiments can be split into three main portions. Firstly, the input data used within the experiment is split categorically into training and testing datasets, before being pre-processed using classical pre-processing fucntions. Secondly, classical feature vectors are subsequently encoded onto the quantum circuit using an appropriate encoding methodology, and the circuit routine is executed. Finally, the quantum circuit is measured and the result of measurement is post-processed classically, where a classification result is obtained. A visual overview of the system pipeline can be seen in fig. 4.1.

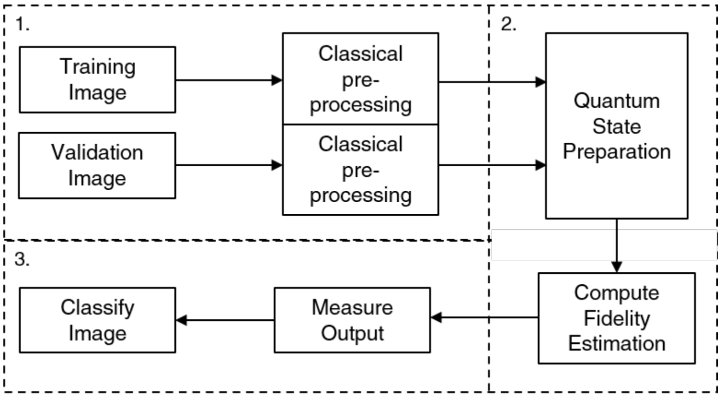


Figure 4.1: A high-level flowchart overview of the implemented system. Firstly, pre-processing is conducted of the image. Then, the image is encoded into a quantum state and the quantum circuit is executed. Finally, post processing occurs and a classification result is obtained.

Data Pre-Processing & Feature Encoding

For the purposes of this experiment, the AT&T face dataset was used. This dataset consists of 400 images of faces, including a variety of subjects with images taken using differing photo angles and expressions. On a technical level, each original image is sized at 92x112 pixels, in a greyscale format containing 256 grey levels per pixel.

The dataset was consequently split into two portions using an image split ratio of 3:1. Whilst not used in the conventional sense, I will refer to the larger portion containing 75% of the face images as the 'train dataset', and the other portion as the 'validation dataset'. This ratio was chosen to provide enough representation for the training data, whilst leaving enough images for a varied and unbiased validation set. Furthermore, for the purposes of validation, a bespoke dataset consisting of 400 images collected from various sources of non-facial objects was also included within the validation dataset portion. Once the two datasets had been finalised, all images were resized to a size of 8x8, prior to being flattened to each form a 64-dimensional vector. Due to the nature of quantum embeddings, a 64-dimensional vector can be fully encoded using 6 qubits only. Some examples of the original images used throughout can be seen in fig. 4.2.

Once the datasets had been formed and transformed into their column vector equivalents, an average face vector was created using the training dataset portion. This was done to create a comparison state that would be

Figure 4.2: Examples of the original images merged to form the dataset used throughout the experiment. All images were subsequently resized to 8x8 and flattened to form a singular column vector.

used to classify a result. To do this, average values were taken over each image in the training dataset per data dimension, i.e:

$$x = \frac{1}{N}\sum_{i=0}^{N} T_i \tag{4.3}$$

Where N is the total number of training images used (in this case, 300) and T is the subsequent image vetor. In order to encode the resulting single average vector into an equivalent quantum state vector, the amplitude encoding scheme was used. For the input size of $D = 64$ dimensions, this would require $\log_2 D = 6$ qubits per input state. Following the amplitude encoding scheme, the input vector $\mathcal{X}$ has to be normalized to produce a valid quantum state $|\psi\rangle$, i.e:

$$\mathcal{X} = \frac{x}{||x||} \quad \longrightarrow \quad |\psi\rangle = \sum_{i=0}^{D} \mathcal{X}\,|i\rangle \tag{4.4}$$

Now that the image vectors are in their valid quantum forms, two quantum states $|\psi\rangle$ and $|\phi\rangle$ can be created using the single average training vector and a single image from the validation data respectively. Once amplitude encoding has been performed on both input state vectors, the system consisting of 13 qubits (6 qubits per input vector + 1 ancilliary qubit) is in the following state:

$$|\mathcal{S}\rangle = |0\rangle \otimes |\psi\rangle \otimes |\phi\rangle \tag{4.5}$$

Circuit Design

After the input state vectors $|\psi\rangle$ and $|\phi\rangle$ have been produced from the training and validation datasets, the quantum circuit is initialized to the state shown in 4.5. From here, a comparison between $|\psi\rangle$ and $|\phi\rangle$ needs to occur. In order to do this, a quantum subroutine known as a *swap test* is performed.

A swap test computes the fidelity, or inner-products, between two quantum states. By computing the overlap between these states, we can determine how similar they are in nature. The swap test is the most computationally expensive portion of the circuit, and requires an ancilliary qubit which contains the result.

To perform a swap test, firstly a Hadamard operator is applied to the ancilliary qubit initialized in the $|0\rangle$ state. This places the qubit into an equal superposition of the computational basis states:

$$|0\rangle \quad \longrightarrow \quad \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{4.6}$$

After this, successive C-SWAP operations are performed between pairs of qubits (with the ancilliary qubit as the control qubit and 1 qubit from $|\psi\rangle$ and $|\phi\rangle$ being the target qubits), until a C-SWAP operation has been performed to all pairs of qubits. The final stage of the swap test routine is the application of a second Hadamard operation to the ancilliary qubit.

Once the swap test routine has been performed, the state of the system is:

$$|\mathcal{S}\rangle \;=\; \frac{1}{\sqrt{2}}(|0\rangle\,|\psi\rangle\,|\phi\rangle + |1\rangle\,|\psi\rangle\,|\phi\rangle) \tag{4.7}$$

As stated previously, the number of qubits required to replicate this experiment is 13, for 6 qubits per input state and 1 ancilla. In order to generalise the system for cases where the input dimensionality may vary, the total number of qubits Q required for a system using D-dimensional datapoints naturally would be $Q = 2\log_2 D + 1$.

Post-Processing & Classification

Once both of the input states $|\psi\rangle$ and $|\phi\rangle$ have been encoded and a swap test has been performed, the output of the circuit must be measured to give a value to post-process and subsequently produce a classification result. Within this system, only the ancilliary qubit is measured, rather than each individual qubit. During measurement, an expectation value with respect to the Z-basis is taken to give the fidelity of the two-encoded states.

After the second Hadamard operation is performed, the measurement output of the ancilliary qubit is as follows, where $P(0)$ is the probability of the qubit measuring as $|0\rangle$:

$$P(0) = \frac{1}{2} + \frac{1}{2} |\langle\psi|\phi\rangle|^2 \tag{4.8}$$

If the two comparison states $|\phi\rangle$ and $|\psi\rangle$ are orthogonal to each other, $|\langle\psi|\phi\rangle|^2 = 0$. If the two comparison states are equal, then $|\langle\psi|\phi\rangle|^2 = 1$. Once the ancilliary qubit has been measured and this probability value is outputted, a classification result can be determined using a threshold value.

If the measured value is higher than the threshold value set, then the result is that the comparison image is that of an individuals face. If the measured value is lower than the threshold, then the result is that it is a non-face image. In order to determine an effective threshold value, repeated experiments are conducted using various threshold values, where the threshold that results in the highest accuracy value is considered optimal.

A visual diagram of the circuit used to implement the methodology described is shown in fig. 4.3.

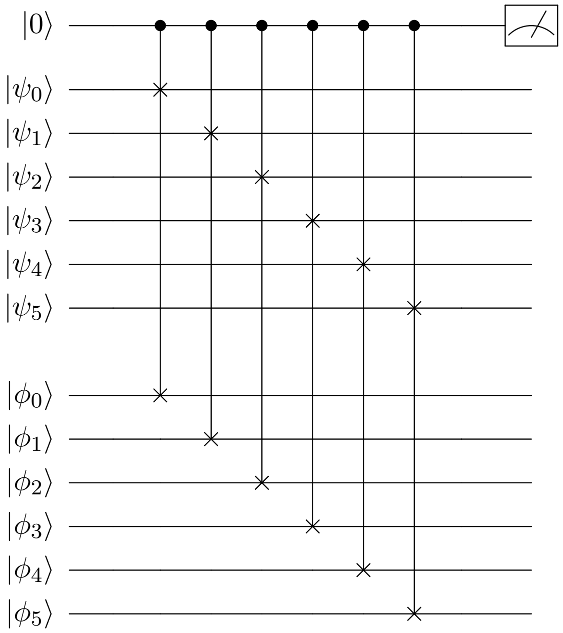


Figure 4.3: Circuit diagram of the system implemented. Two images are encoded onto states of $|\psi\rangle$ and $|\phi\rangle$, where a swap test routine is executed prior to measurement of an ancilla qubit.

4.3 Results

4.3.1 Quantum Facial Identification Results

In order to demonstrate the potential of the system and produce a high accuracy in classification performance, there has to be a substantial difference to

No. of Qubits	Input Vector Dimensions	Average Fidelity of Face Images	Average Fidelity of Non-Face Images
9	16	0.95434	0.87156
13	64	0.96199	0.82020
17	256	0.86848	0.78061

Table 4.1: Average fidelity values for face/non-face images per varied input image sizes.

the fidelity values of face and non-face images. To provide a deeper analysis into the performance of the system, classification performance results were taken using various input image dimensionalities, and subsequently various numbers of qubits.

For the purposes of these experiments, the system was developed using the quantum software library PennyLane [135], and simulated using IBM's high-performance simulators [136].

Table 4.1 shows values for the average fidelity of images containing a face and images that do not contain a face. Various input image sizes are considered, which subsequently affect the total number of qubits used in the system and the runtime speed. The maximum number of qubits tested was 17, as it was felt that a much larger and complex system size is beyond the scope of the experiment and was unecessary.

Here, it can be seen in table 4.1 that a smaller scale of input with 9 total qubits produced similar fidelity values, whereas a larger input dimensionality using 13 qubits produced more disparity between fidelity values. However, the largest input size tested using 17 qubits produced a similarity close to that of the 9-qubit system.

Overall, these average fidelities suggest that there is a balance that needs to be considered when designing quantum circuits between the number of qubits used and the corresponding output. For the smaller system size of 9 qubits and 16 input image dimensions, we are losing important detail required to determine whether the image is that of a face or not due to the lack of pixels involved. Whereas, for the largest system size of 17 qubits, there is too much variability involved in the larger quantum states compared, thus causing the respective fidelities to drop, as well as becoming more similar.

To determine the classification performance of the system, various threshold levels were considered to maximise accuracy of classification on the test dataset consisting of 100 face and 300 non-face images. The threshold values considered started at a value of 0.7, and were incremented in steps of 0.01 until a final value of 1.0 was reached. A starting threshold of 0.7 was

Table 4.2: Accuracy and the best performing threshold value per system size used.

No. of Qubits	System Accuracy	Highest Performing Threshold
9	0.800	0.971
13	0.906	0.958
17	0.800	0.993

chosen as a smaller threshold would have been too ambiguous to distinguish between face and non-face images. The best performing results from these tests are shown in table 4.2.

In fig. 4.4, the relationship between classification accuracy on the test dataset and the threshold value used can be seen for the 13-qubit system. Naturally, there will be a single peak in the curve, where the similarity cut-off would be to determine whether an image is that of a face or not.

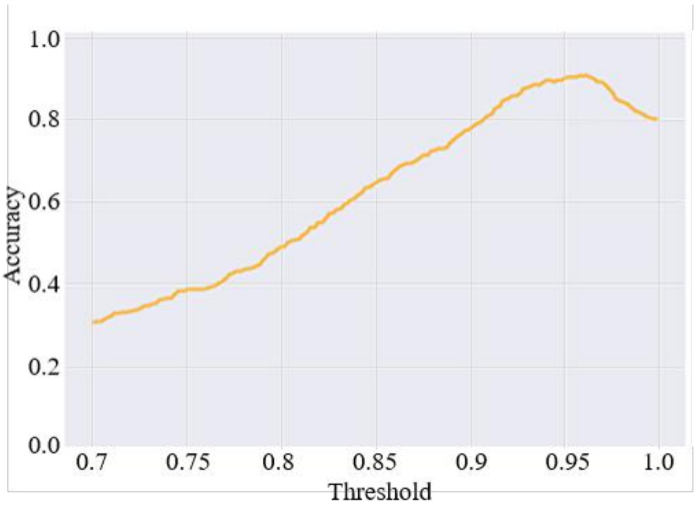


Figure 4.4: Curve showing the relationship between the test-set classification accuracy per threshold value for the 13-qubit system.

4.3.2 Comparitive Classical Algorithm Results

In order to demonstrate the performance potential of the quantum system developed, it is important to give some context as to what a realistic result to expect would be. To do this, the results gathered were compared against results gathered using the classical machine learning algorithms of an SVM, as well as a k-NN. These algorithms were chosen as they are tried and tested methodologies, which have been commonly used in the field of facial identification [137, 138].

Table 4.3: Accuracy comparison between the proposed quantum system versus classical SVM and k-NN algorithms examined.

Algorithm	Test Dataset Accuracy
SVM	0.995
k-NN (k=2)	0.945
Quantum	0.906

For the purposes of the following experiments, the previous datasets used were merged, before being split 75% training data to 25% testing data. In total, this amounted to 300 training images per class, and 100 testing images per class of face or non-face. Algorithm specific, the SVM used consisted of 155 support vectors and used a Gaussian kernel. For the k-NN algorithms, various k-neighbour values were chosen, ranging between 1-20.

From the results in table 4.3, it can be seen that the classical algorithms do perform to a slightly better level of accuracy that that of the proposed quantum methodology. However, it could be argued that the proposed methodology does still perform to a fair performance level. As shown in fig. 4.5, classification accuracy for various values of k also remain within $90 - 95\%$ boundary.

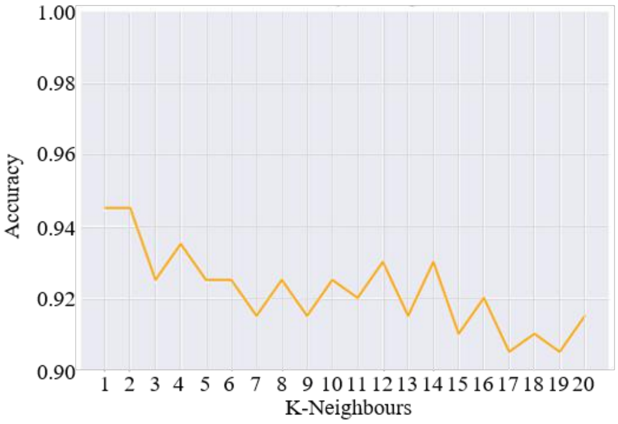


Figure 4.5: Classification accuracy on the test dataset portion for the k-NN algorithm used, with varied k-neighbour values.

4.3.3 Computational Complexity Analysis

Whilst the early results presented previously are promising towards the use of quantum computing for facial identification purposes, a key aspect which should be considered is the complexity of any algorithm created. A key aspect driving the development of quantum algorithms for machine learn-

Table 4.4: Computational complexity comparison between quantum and classical algorithms used throughout these experiments.

Algorithm	Complexity
SVM	$O(I^2D + I^3)$
k-NN (k=2)	$O(DI)$
Quantum	$O(I \log_2(D))$

ing purposes is the potential benefit to the computational complexity in comparison to current classical methodologies. In order to display a fair comparison of the showcased quantum method to classical algorithms, the computational complexity must also be compared alongside the previous accuracy metric.

Table 4.4 shows a comparison of the relevant computational complexities for each methodology used throughout this experiment, where D is the dimensionality of each input vector and I is the total number of images used. Here, it can be seen that the proposed quantum method achieves exponential speedups in areas over the compared classical methods analysed. This is due to the efficient representation of quantum information from the nature of the quantum embedding used (amplitude embedding being $\log_2$ naturally).

4.4 Discussion

From the experimental results explored, the quantum algorithm developed within this chapter classified the data to a slightly lower initial level of accuracy compared to the classical ML algorithms of SVM and k-NN. However, there was a certain degree of optimization occuring within the classical methods. For the SVM, the margin of seperation for the seperating hyperplane is maximised, and for the k-NN the optimal k-neighbor value within the allocated range is located. Whilst the optimum threshold was found for the quantum algorithm, this is most similar to the optimization of the k-NN (where accuracy decreased rapidly after k= 2) and is not of the same level of complexity as the SVM. Therefore, it could be argued that comparison against the SVM was unfair but regardless a good indication of realistic classification performance expectations from the dataset was provided. Once computational complexity of each algorithm is considered, the speedup potential benefit of quantum computing via efficient quantum information representation is realised.

The experimental results obtained in this research suggest that the designing of quantum algorithms should be balanced in the sense of detail and variability. This was supported by results in table 4.1 which showed the middle system size to have the highest classification performance. Therefore, we should aim to utilise enough qubits to create a sufficiently complex feature space that can contain a high level of information. However, the amount of qubits used should not become too many to the point of creating an uncontrollable level of variability within the contained quantum state, hence limiting our output measurement potential.

Whilst the early experimental results gathered for the quantum algorithm are promising, they are not without their limitations. Additional studies should be conducted into the robustness of the algorithm itself in relation to fidelity estimations and qubit cohesion in NISQ environments. Alongside this, classification performance should be analysed when a variety of image perspecives are included, such as varied ratios and lighting settings. Whilst the non-facial data did include these aspects, the facial training data was of a limited format. Therefore, the system may struggle to determine between a face and similarly shaped objects, subsequently affecting the robustness of the methodology.

Alternatively, other methods that could be considered include altering the pre-processing portion of classification. Here, various pre-processing steps, or different methods of qubit encoding could be explored. Different methods of quantum information encoding in particular may be a promising avenue for further investigation. Whilst Amplitude encoding is fairly efficient in nature, requiring $\log_2$ qubits for state dimensions D, different schematics may create a more efficient or enriched feature space that can limit the problem of qubit variability described.

As a whole, the results achieved by the quantum computing algorithm explored are promising in comparison to the classical ML methods tested, but only once computational complexity is considered. Although there exists quantum systems large enough to process the number of qubits used in these experiments, it is questionable whether the functionality of these qubits will be at a cohesive enough level for the system to function at a good performance rate. Once improvements are made to quantum hardware, specifically towards qubit cohesion, methods such as the one explored in this chapter may become much more realistic and feasible to use as part of real-world application domains.

4.5 Conclusion

In this chapter, the concept of quantum computing applied to the widely-used machine learning domain of facial identification was explored. By performing Amplitude encoding, high-dimensional feature vectors could be encoded into their equivalent quantum state, creating a highly-complex Hilbert space. Using a quantum SWAP test subroutine, the fidelity between two quantum states could be measured effectively. An image could then be classified appropriately as either a face or non-face in conjunction with a threshold function.

Initial experimental results of the developed system show promise which can only hope to be improved upon with further investigation and development. Whilst the examined classical algorithms of SVM and k-NN classified the dataset to a higher standard than the quantum algorithm, the computational compelxity of the quantum algorithm outperformed them both. By efficient representation of quantum information, speedups of exponential scale could be achieved, with a slight trade-off of decreased classification accuracy. Considering the NISQ era of quantum computing, this trade-off could be considered fair and justified. As advances to quantum hardware are made over time, it is hoped that the initial performance benchmarks produced here can be improved, and that the real-world applicability of the algorithm could be realized.

Finally, it is suggested that alternative methods of encoding should be investigated in order to expand upon the findings of this chapter. This should be done to determine whether rich feature spaces can be created in a qubit-efficient manner of encoding to minimize the limitations discussed within this chapter.

Chapter 5

On Depth, Robustness And Performance Of Classification Via Single-Qubit Encoding

5.1 Introduction

The following chapter presents an investigation into the capability of single-qubit encoding, a method proposed for information encoding and classification using a single qubit only. This is conducted following research presented in chapter 4 in an effort to answer the following research question:

Research Question 3: *Can effective feature representations of classical data be created in Hilbert space using low numbers of quantum bits?*

VQCs are currently an area of large interest within QML [12, 36, 69, 72, 88, 115, 139–141] and provide a natural progression point for the development of QML algorithms due to their optimization capability. VQCs often appear to be initialised using circuit structures and designs that are seemingly chosen at random, or have very little justification as to why [142]. Whilst this may work fine in certain scenarios, we need to look at what aspects of these circuits improve classification performance, and whether specific features such as depth to VQCs, are most beneficial to doing do.

Two measures of circuit capability, referred to as 'expressability' and 'entangling capability' were initially explored in [142]. This was furthered in [143], where the resulting performance of these circuits were compared in a classification task setting. These studies ultimately suggest that expressability and classification performance of VQCs will begin to plateau at a point, however this point is not absolute and may change dependent to the circuit used. A particular encoding and classification strategy that has shown to be promising for an individual qubit is the concept of data re-uploading, introduced in [125]. Here, layers of parametrized unitary gates are repeated to embed classical input data into Hilbert space. As a minimum, only a single qubit is required for classification, making the promising methodology appealing to pursue further as a whole.

A critical aspect that should be explored when using data re-uploading is the correlation between circuit parameters and performance. A parameter here could be considered as system depth (i.e. the number of uploading layers used). The original proposal of data re-uploading partially explored this parameter, where added depth to the circuit did show a subsequent increase to classification performance prior to saturating. However, arguably there was not enough evidence in their results to support that increasing circuit depth, or other parameters they varied such as the number of qubits, or the use of entangling layers is always necessary to consistently improve upon classification performance.

Many QML algorithms are designed and tested within a noise-less simulated environment. Whilst these simulations can be effective in determining optimal performance, they leave an important factor out of how the experimental results of the proposed system may translate across to a real-world task through a QPU where environmental noise is present. An analysis in [144] took this into account, showcasing results processed using a QPU. However, little insight was provided into showing any correlation between circuit parameters and performance.

Ultimately, effective use of each qubit is especially important at the current NISQ era of quantum computation, as we are fairly limited by qubit cohesion and connectivity in QPUs. Many quantum algorithms rely on moderate to large numbers of cohesive qubits to compute or encode input data. This is not necessarily a practical method to use at the current time, which the results displayed within chapter 4 highlight. Alongside this, it is especially important that we can maximise the working potential of each qubit used during computation, so that if a methodology is extended to multiple qubits, this is done in an efficient manner. Doing so will not only allow us to understand the computational power that a single qubit possesses, but also provide valuable insight into effective VQC design where each qubit can be maximally used. Because of the reasons outlined here, this research focuses on the use and application of a single qubit only.

Overall, there are important aspects that have not been necessarily examined in recent works that explore data re-uploading [143, 144]. This includes the investigation of the correlation between system parameters and classification performance (using classification metrics of loss and accuracy), as well as how the influence of envrionmental noise affects said performance. These are aspects which should be examined together in order to gain a full understanding of the methodology, as well as how this may translate to the wider field. Therefore, the aim of this chapter is to determine any correlations present between circuit parameters and classification performance, as well as determine how this performance may translate to application within noisy quantum environments, using a single qubit only.

Ultimately, the contribution of this research is through an analysis of classification performance using the data re-uploading single-qubit classifier. The aim of the experimental results in this chapter are to identify key trends within system design which can not only aid classification performance, but also improve robustness of training within noisy environments. The research presented here will improve our understanding of performance using the data re-uploading methodology, as well as how we can adapt VQC design to maximise the effectiveness of each available qubit, dependent to the environmental noise levels present.

In order to achieve these aims, previous work will be bridged through an analysis of classification performance using varied circuit depths, applied to artificially-generated datasets of incrementing difficulty. The resulting embeddings will be examined where necessary, to give an indication of how they will change dependent to the input and design of the VQC. This will aid the search in determining effective embeddings of classical data, that produce enriched feature spaces capable of higher-performing standards of classification. In addition, this will determine whether the methodology remains viable as the dimensionality of the task increases simultaneously.

Alongside this, a case study will be conducted using MNIST data to provide a realistic indication of how the methodology may translate across to a scenario where non-artificial data is being used. The inclusion of this case study will help to negate any biases that may have unknowingly occurred due to the inclusion of artificially-generated datasets. In addition to the previous points, the methodology will be tested using a simulated noisy quantum environment. Doing so will help identify any design considerations that may assist convergence during training, thus allowing for higher levels of classification accuracy to be achieved in a shorter timeframe.

The findings of this research relate to an in-depth analysis of the data re-uploading methodology. In this chapter, any correlation between increased system depth supporting improved classification performance is identified and quantified. The experimental results gathered and analysis performed also supports that increasing system depth can boost stability during training in quantum environments where environmental noise is present, leading to better overall classification performance.

The structure of this chapter is as follows. Firstly, the methodology of data re-uploading is briefly introduced to provide the background knowledge required. Then, an outline of the experimental setup and configuration will be described, followed by the presentation of the experimental results. Afterwards, an analysis of the experimental results will be conducted, where key aspects and trends can be identified in order to form any conclusions and propose areas for further investigation or development.

5.2 Methodology

Within the field of Machine Learning, data is often presented, or transformed to the form of a column vector prior to being processed. Naturally, this ties in favourably with quantum computing, which by nature deals with column vectors primarily. Data re-uploading is a methodology in which we can encode these vectors and create a feature Hilbert space using successive unitary operations that act on each dimension of the input. For any $SU(2)$ operation, U (i.e. a 2×2 unitary matrix with determinant 1), we are able to decompose the operation into the following [145]:

$$U = e^{i\alpha} R_Z(\beta) R_Y(\gamma) R_Z(\delta) \tag{5.1}$$

Where $\alpha \in \mathbb{R}$ is the global phase factor and $\beta, \gamma, \delta \in \mathbb{R}$ are the Euler angles that define each rotation around the Z, Y and Z axis respectively. From here, we can define the Euler angles as:

$$\beta = \theta_i + x_i \cdot \phi_i \tag{5.2}$$

$$\gamma = \theta_{i+1} + x_{i+1} \cdot \phi_{i+1} \tag{5.3}$$

$$\delta = \theta_{i+2} + x_{i+2} \cdot \phi_{i+2} \tag{5.4}$$

With θ and ϕ being weight parameters fed into our optimization protocol, and x_i being the value of the input x at dimension i. these trainable weights define the extent to which the state of the qubit is rotated, with respect to the value of the input. From these parameter definitions, we can utilize a maximum of three input dimensions per unitary operation performed.

From here, the input vector can be cycles through, encoding a set of three data dimension values at a time until the input vector has been fully encoded. In the proposed methodology, this is referred to as a full 'upload layer' of the data. By repeating this embedding of input data and adding successive uploading layers, a highly-complex feature Hilbert space can be created in an attempt to improve the learning capacity of the algorithm, and thus boost classification performance.

After the input vector has been uploaded to the specified number of times, then the fidelity of the encoded quantum state can be measured with respect to a target state. For each task, a set of target states are chosen which are maximally distanced for each other. As an example, for a binary classification task, these target states could be configured to be state $|0\rangle$ for each datapoint contained within class 0, and $|1\rangle$ for each datapoint contained within class 1.

The loss function that is used throughout this research is based on the weighted fidelity loss function defined in [125], however we exclude the indvidual class weightings. Defined in eq. 5.5, the loss function aims to minimize the fidelity (distance) F of datapoints between their encoded state $F(\vec{\theta}, \vec{x_d}, \vec{\phi})$ and their respective target states $F_c(\vec{x_d})$, where θ and ϕ are parameters to be optimized, x is the input data, d is the datapoint from training/validation dataset of size D and c is the current class of total number of classes used C.

$$\frac{1}{2}\sum_{d=1}^{D}\sum_{c=1}^{C}\left(F(\vec{\theta}, \vec{x_d}, \vec{\phi}) - F_c(\vec{x_d})\right)^2 \tag{5.5}$$

A full detailed description of the data re-uploading methodology can be found in its proposal in [125], however, for simplicity an abstract overview can be seen in fig. 5.1.

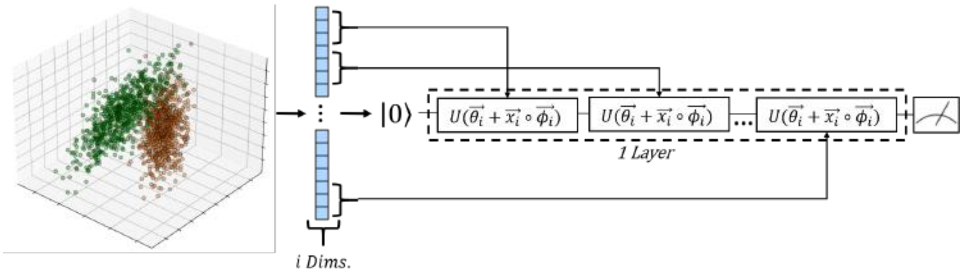

|0⟩
$U(\vec{\theta_i} + \vec{x_i} \circ \vec{\phi_i})$
$U(\vec{\theta_i} + \vec{x_i} \circ \vec{\phi_i})$
...
$U(\vec{\theta_i} + \vec{x_i} \circ \vec{\phi_i})$
1 Layer
i Dims.

Figure 5.1: An abstract overview of the data re-uploading process from i-dimensional datapoint to measurement. Firstly, the input datapoint is recognized as a single column vector x. Then, each input dimension is 'uploaded' by an arbitray unitary gate U, which uses a weighted sum of 2 rotational parameters (θ, ϕ) per dimension. This process is repeated until each datapoint dimension has been encoded, where the qubit is finally measured with respect to a target state.

5.3 Results

5.3.1 Experimental Setup

Barren Plateau Problem

As a preface to the description of the following experiments, it is relevant to address the barren plateau problem that is largely present when training VQCs, as well as the effect that this has if not considered. Barren plateaus are areas of near-zero gradient within the loss landscape that, if not considered, can substantially affect the training of VQCs and not allow for stable convergence to a satisfactory local minimum in a sufficient number of epochs.

The problem of barren platueas was first addressed in [146], where several approaches have since been considered to avoid or minimize the effect of this problem, such as local cost functions [147, 148], evaluating initialized weights [149], or the use of quantum natural gradient [150, 151].

For the following experiments, the problem of barren plateaus is considered by initializing 10 randomized weight sets, where all weight sets are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. These values were chosen from results gathered in preliminary experiments, as they produced more consistency between training samples compared to other weight initialization values considered.

For each weight set, a single epoch was conducted on the test dataset portion. The parameter set that corresponded to the lowest test set loss value in this initial epoch was then used thereafter for the remainder of the training for that experiment. Whilst this helped to avoid the problem of barren platueas in our experiments, it should be acknowledged that this was a temporary solution to the problem only, and alternative measures should be analyzed for a capable long-term solution to the barren platuea problem.

Experimentation Plan

To outline the following results, the concepts targeted within these experiments will be addressed in order. Firstly, to consider how depth affects classification performance of the data re-uploading scheme, the number of layers used within the system will be incremented from 1 to 10. Alongside this, the dimensionality of the input data will be varied, starting from 3 dimensions up to 15 in increments of 3. This simulates increased difficulty of the classification task, which will determine whether any trends remain consistent when this difficulty is increased. In addition, this will also be extended from binary classification tasks to multi-class classification tasks (with 3 classes), in order to provide reasonable assumption to how this performance may translate to other potential tasks.

Secondly, within QPUs, quantum noise from external factors (e.g. influence from the environment) can negatively impact the output of the quantum system executed. To account for this, simulated noise will be introduced, with varying levels of noise strength, in order to determine the robustness of the system in comparison to the amount of depth implemented.

Thirdly, to account for any bias that may have been created from the use of synthetic datasets until this point, a case study consisting of a subset of the MNIST dataset will be analysed. This will also give a fair indication of the performance that the methodology may bring when used towards a real-world task scenario.

For all of the following experiments, the PennyLane library [135] in conjunction with the PyTorch optimization protocol was used to develop the experimental framework. For non-noisy simulation environments, the Qulacs [152] qubit simulator external plugin was used within the PennyLane interface. For noisy simulation environments, the mixed-state simulator native to the PennyLane library was implemented. For reproducibility purposes, all relevant randomization seeds were set to zero, unless otherwise stated.

For all artificially-generated datasets, these were generated natively within the scikit-learn library [153]. This allowed for much greater flexibility in defining the data and feature creation parameters used appropriately. For each dataset, unless otherwise states, these creation parameters were set to include 2 informative features, 1 redundant feature, 1 cluster per class, a class separation of 1 and a random generation seed of 1234.

5.3.2 Layer Correlation Analysis

For the following results, artificial datasets were generate, consisting of 500 train and 1500 test images split evenly between the number of classes used. 30 epochs of training were conducted per experiment, using the SGD optimization protocol with a learning rate of $10^{-}2$, unless otherwise stated. These hyperparameters were chosen from conducting preliminary experiments, where more stability was seen on average during convergence in comparison to higher learning rates, whilst also reducing the amount of optimization epochs needed to converge in comparison to lower rates of learning.

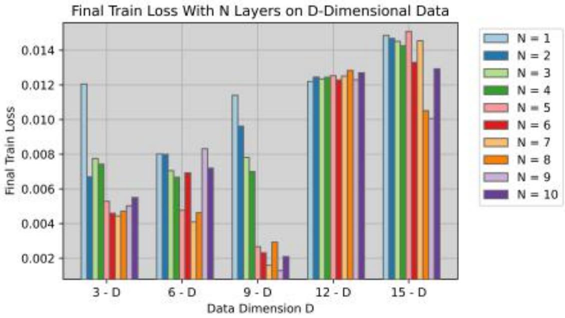


Figure 5.2: Charts displaying training-set loss results of a binary classification task, using layer depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

Fig. 5.2 to 5.5 displays results of a binary classification task, where the depth of the system (i.e. number of layers) is varied from 1 to 10. The dimensionality of the dataset used is also increased from 3 dimensions to 15 dimensions, in intervals of 3. The generally expected behaviour here would be for the overvall trend of classification performance per layer over each dimension to worsen, due to the scaling of difficulty of the task, with each individual dimensional groups' performance improving as layers are added, giving increased learning capacity to the system.

The chart displaying test set loss in fig. 5.3 is fairly consistent with this behaviour until 12 dimensions are reached. From here, the behaviour almost reverses, where the performance of the system does not improve with

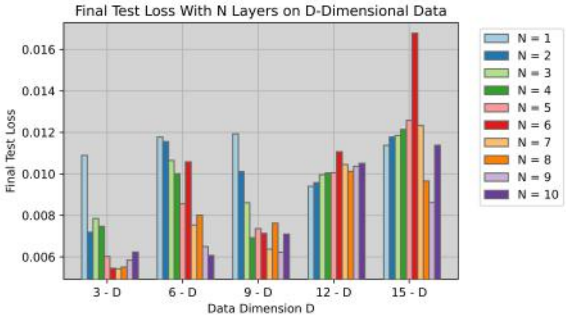


Figure 5.3: Charts displaying testing-set loss results of a binary classification task, using layer depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

additional depth until after 6 layers, where it appears to plateau.

Focusing on 3-dimensional loss results, fairly consistent performance increases can be seen with added depth until 7 layers, where performance starts to regress and worsen thereafter. This behaviour is not unique and happens on more than one occasion. Regardless of the slight regression within the loss value, the system still classifies the vast majority of the test dataset correctly, and does not change throughout the different depth values implemented.

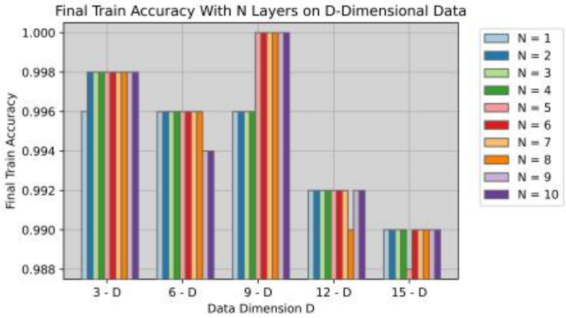


Figure 5.4: Charts displaying training-set accuracy results of a binary classification task, using layer depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

To follow with the loss values displayed in figures 5.2 and 5.3, figure 5.4 displays the classification accuracy values for the training data on the final epoch. Whilst there is some suggesion of improvement for 3-dimensional and 9-dimensional data as depth is increased, this is also contrasted with 6-dimensional data which loses accuracy towards the upper depth values. However, with this said, the classification accuracy for all of these experiments was high regardless, with very minor changes if any. Therefore, it may be of more value to focus the analysis on loss rather than accuracy for this set of results due to the loss equating to confidence in the classification.

As with figure 5.4, there is no clear correlation between depth and classification accuracy on the testing set of data. Whilst there exists some evidence

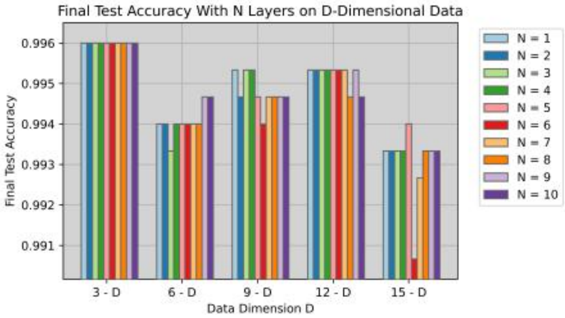


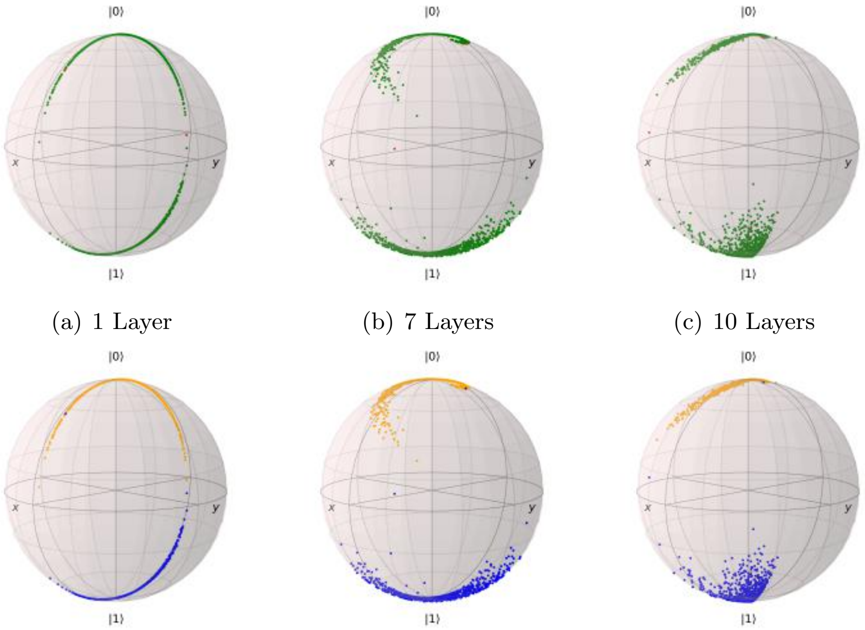
Figure 5.5: Charts displaying testing-set accuracys results of a binary classification task, using layer depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

to implicate a performance improvement, such as 6-dimensional data, there is also evidence to contrast this, such as with 9-dimensional data. Therefore, it may also be of more value to focus primarily on loss results for this set of testing data to provide more insightful analysis.

In order to classify a datapoint, the target state that the datapoint is closest to (i.e. has the highest fidelity value towards) will become its resulting class. For a binary classification task, these target states are located on opposing spaces of the bloch sphere, where states $|0\rangle$ and $|1\rangle$ have been used for simplicity. With this knowledge, two distinct distributions of datapoints (one cluster per class) should appear within each hemisphere of the Bloch sphere. As the cost function used aims to minimize the fidelity between datapoints and their respective target classes, the distributions seen should become progressively smaller and more defined as the loss value decreases during training. As this scheme is extended to multiple classes, the target states become a set of maximally-spaced state vectors, where the same principles are followed regarding the highest state fidelity is the associated class for each datapoint. In essence, this will mean that the respective class boundaries will become reduced in size as more classes are included for a single qubit.

Figure 5.6 displays embeddings of test set data visualized using the Bloch sphere, with layer depths of 1 (subfigures 5.6(a) and 5.6(d)), 7 (subfigures 5.6(b) and 5.6(e)) and 10 (subfigures 5.6(c) and 5.6(f)) at epoch 30. Regardless of the number of layers used for these experiments, it can be seen that the system still classifies the vast majority of datapoints correctly (green point equals a correct classification result), minus a few clear outliers that are heavily nested inside the opposing class cluster.

Whilst the majority of datapoints were classified correctly in figure 5.6 regardless of system depth, an advantage of increased depth shown is for the increased complexity of embedding. In this case, a system depth of a single layer produced a linear embedding, where the distribution of datapoints is narrow along a particular rotational line. As the system depth is increased



(a) 1 Layer (b) 7 Layers (c) 10 Layers

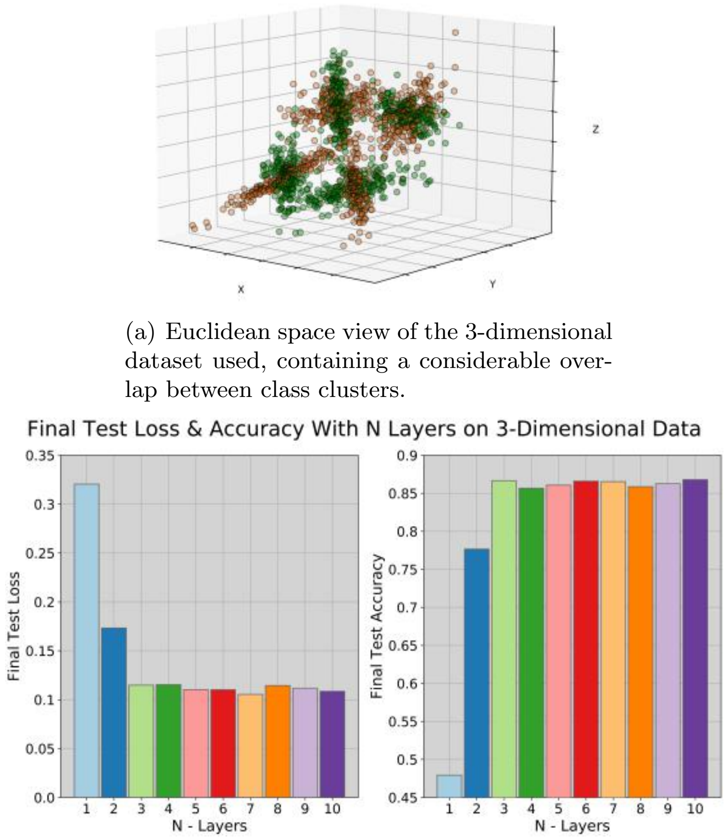
(d) 1 Layer (e) 7 Layers (f) 10 Layers

Figure 5.6: Bloch sphere visualizations of test set embeddings at epoch 30 with varied system depth. Subfigures 5.6(a), 5.6(b) and 5.6(c) show correctly classified points (green) versus incorrectly classified points (red). Subfigures 5.6(d), 5.6(e) and 5.6(f) show the distribution of classes, using a colour to indicate a target class value.

up to 10 layers, the distribution of datapoints becomes much more rotated and distorted. In this case, this allowed for rotations to occur in the test set embeddings, thus forming a more complex feature space. This is highlighted more easily in subfigures 5.6(d), 5.6(e) and 5.6(f)

In the case of the dataset used for this experiment, it could be argued that only a simpler complexity of embedding was required to seperate the datapoints into their respective class boundaries and classify them to a high degree of accuracy. However, for a dataset that is not so seperable with an intense overlap between class clusters of datapoints, the advantage brought by increased system depth may become much more apparent.

Figure 5.7 displays results of a binary classification task using the secondary dataset that contains an extreme overlap between class datapoint clusters, more so than the data used previously for the experimental results shown in figures 5.2 - 5.5 and 5.6. This dataset, also displayed in figure 5.7, was generated using creation parameters of 3 informative features, 4 clusters and a class seperation of 2, all defined within sci-kit learn. In total, this dataset consists of 500 training set and 1500 test set images, split evenly between the 2 classes. Each experiment was trained for 10 total epochs, using



(a) Euclidean space view of the 3-dimensional dataset used, containing a considerable overlap between class clusters.

(b) Charts displaying test set loss (left) and test set accuracy (right) at epoch 10, with varied system depth between 1-10 layers.

Figure 5.7: Figure displaying a visual distribution of the second dataset and associated experimental results.

SGD optimization protocol and a manually tuned learning rate of 10^{-3}, to avoid overfitting in this case.

As previously seen, improvements in classification performance occur as the system depth is increased. However, these improvements are sharpest within the first few layers, and any improvement appears to saturate after approximately 3 layers are used. In comparison to previous results, these increases in classificatino performance with additional depth are at a smaller scale. In addition, there are a much higher proportion of misclassified points using a lower number of layers, when compared to previous experimental results gathered and displayed in figures 5.2 - 5.5.

If we look to the embeddings of the test set data used, displayed in figure 5.8, the embedding capability of a single uploading layer is much more rigid and restricted in comparison to a depth of 10 layers, which allows for a greater degree of flexibility in its' mapping of datapoints. This greater

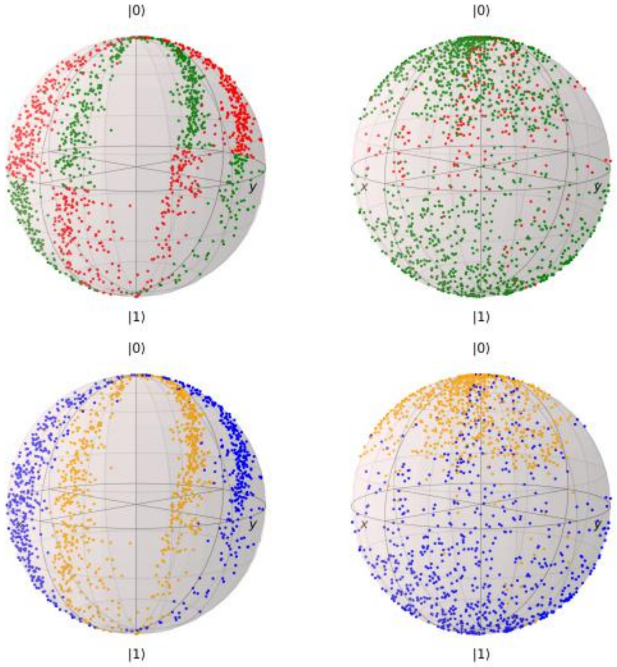


Figure 5.8: Bloch sphere visualizations displaying embedding of the test set datapoints associated with the dataset displayed in figure 5.7. The top row displays correctly classified points (green) and incorrectly classified points (red). The bottom row displays the distribution of individual classes (yellow and blue coloured points corresponding to class states $|0\rangle$ and $|1\rangle$ respectively). Left to right on both rows displays results using a system depth of 1 and 10 layers respectively.

embedding flexibility results in the system being able to seperate each class much more effectively than with a lower embedding flexibility.

In the case of the experimental results displayed in figures 5.2 - 5.5, a system depth of a single layer, requiring 6 parameters in total, was sufficient to classify the dataset to a high standard. Whilst increasing system depth generally improved the confidence of these scores up until 7 layers, the additional layers were unecessary to determine much better performance, and ultimately just increased compelxity.

For the secondary 3-dimensional dataset with a severe overlapping between class data clusters (displayed in figure 5.7), a much more complex level of embedding was needed to classify the dataset to a high standard. This was not sufficiently found until a system depth of 3 layers and onwards, where the performance increases between layers 1, 2 and 3 for results in figure 5.7 are also much larger in proportion to those shown previously in figures 5.2 - 5.5.

As mentioned previously, when the nature of the task is extended to multiple classes for a single qubit, the respective class boundaries will become smaller as the number of classes used grows larger. Because of this, for a multi-class classification task, higher levels of embedding flexibility may be required to efectively seperate and map each datapoint into their respective class region, more so than a binary classification task.

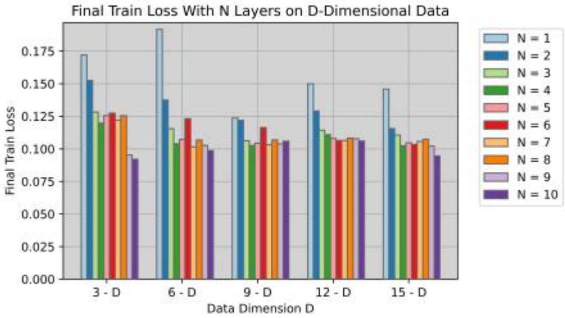


Figure 5.9: Training-set loss results of a 3-class classification task, using varied system depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

Figures 5.9 - 5.12 display 4 charts with results gathered from a 3-class classification task, consisting of the default data generation scheme described earlier in section 5.3.1.

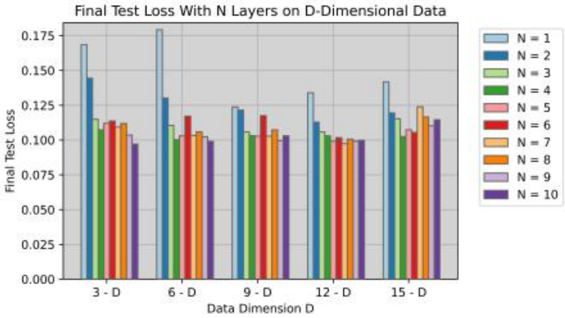


Figure 5.10: Testing-set loss results of a 3-class classification task, using varied system depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

Looking at the general behaviour displayed between depth and data dimensionality in figure 5.9 and 5.10, instantly it can be seen that there is a correlation between system depth and loss values. The correlation shown with these results is also arguably more stronger than that of the initialy gathered set of results displayed in figures 5.2 - 5.5.

Whilst performance improvements can be seen with added system depth, these improvements do saturate and begin to plateau at a point. On average, the sharpest performance improvements occur between 1-3 layers, but quickly plateau thereafter. As seen before, there are cases where performance begins to regress, such as test set performance using 15-dimensional data as an example in figure 5.10. However, as there is a spike at the corresponding train set classification accuracy in figure 5.11, it is unclear whether

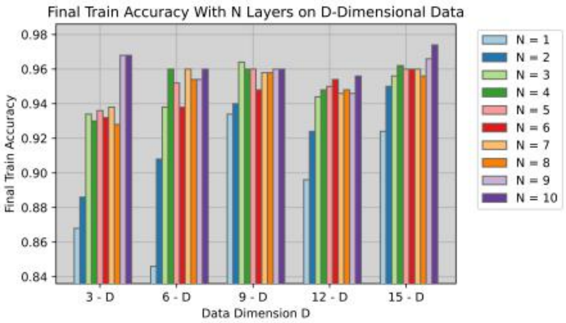


Figure 5.11: Training-set accuracy results of a 3-class classification task, using varied system depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

aspects of these performance regressions are due to a slight overfitting towards the training data.

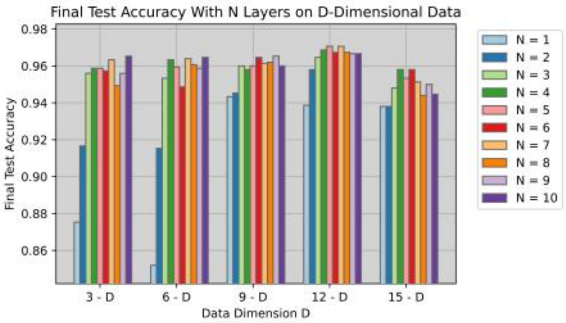
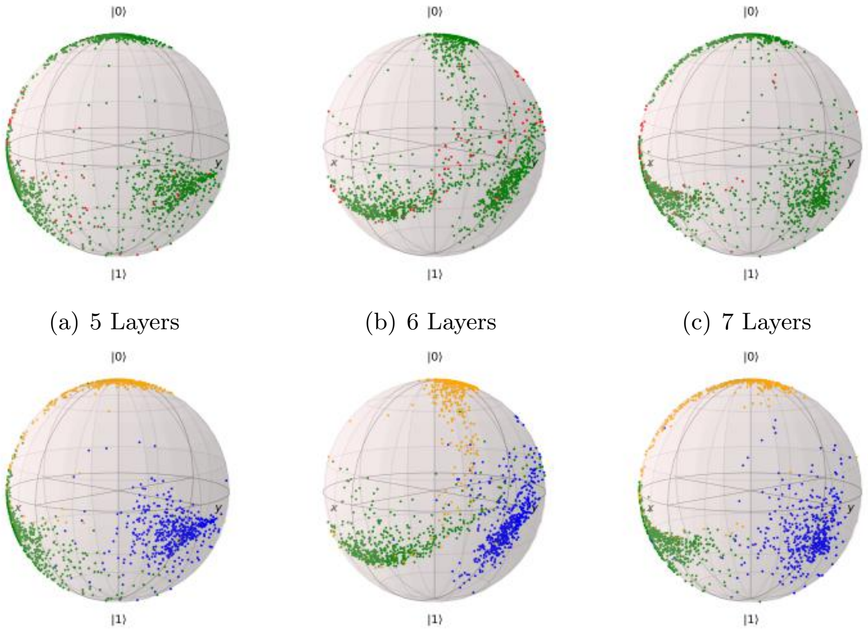


Figure 5.12: Testing-set results of a 3-class classification task, using varied system depths $N = 1..10$ per each dataset dimension $D \in \{3, 6, 9, 12, 15\}$.

In figures 5.2 - 5.5, the classification accuracy using a single layer was much higher in proportion to successive layers than in the case of results shown in figures 5.9 - 5.12. This suggests that the dataset here was much harder to classify to a high standrad, and the complexity of embedding was a key feature in determining classification performance overall.

Upon a closer review of the experimental results gathered, there are cases where test set classification performance begins to regress. This is apparent with 6-dimensional and 9-dimensional dataset results, where a spike occurs when a system depth of 6 layers is used. When inspecting the corresponding embeddings displayed in figure 5.13, it can be seen that the embeddings associated with layers 5 (figure 5.13(d)) and 7 (figure 5.13(f)) are very similar in nature. However, the datapoint distribution formed using 6 layers (figure 5.13(e)) is much closer together, meaning that the datapoints will be further away from their respective target state, thus producing a small increase in loss and a drop in classification accuracy near to where all three class state boundaries intersect. This is evidenced in the more prominent scattering



(a) 5 Layers (b) 6 Layers (c) 7 Layers

(d) 5 Layers (e) 6 Layers (f) 7 Layers

Figure 5.13: Bloch sphere visualizations displaying subsequent test set embeddings of the 6-dimensional dataset used to gather experimental results shown in figures 5.9 - 5.12. Subfigures 5.13(a) - 5.13(c) show visualizations of correctly classified datapoints (green) versus incorrectly classified datapoints (red) at epoch 30. Subfigures 5.13(d) - 5.13(f) show visualizations of the distribution of datapoint embeddings with corresponding class values (one colour per class).

of incorrectly classified points (red in colour) shown for figure 5.13(b) when compared to figures 5.13(a) and 5.13(c).

Figure 5.14 displays test set loss results gathered using a system depth of 5,6 and 7 layers, and the 6-dimensional dataset used in figures 5.9 - 5.12. Here, it can be identified that a system depth of 6 layers has a much slower convergence rate in comparison to system depths of 5 or 7 layers in this case. However, where the loss curves for 5 and 7 layers appear to plateau, the loss curve for 6 layers is still steadily decreasing. This implies that the initial weight distribution for 6 layers may have been initialized in a region of lower gradient than that of the initial weight selection for layers 5 and 7. Subsequently, this would cause a delay in convergence, similar to the behaviour shown in figure 5.14. If system training were to be continued for a number of epochs following this, then it could be argued that the loss curves would meet at roughly the same boundary between loss values of $0.10 - 0.11$.

Whilst weight initialization may be a factor in the drop of performance in this case, it is difficult to state that this factor cause performance drops in other cases throughout the other experiments conducted. As an exam-

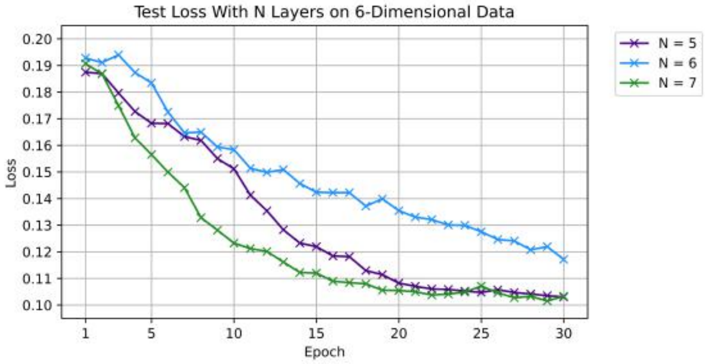


Figure 5.14: Plot showing test set loss results per epoch, using 6-dimensional data and a system depth of 5, 6 and 7 layers.

ple, there are cases of performance regression occuring (i.e. test set performance using 15-dimensional data at 10 layers displayed in figureS 5.9 - 5.12), however, there is also a drop in the corresponding train set classification performance. This suggests that there may have been a slight overfit to the training data, causing a drop in classification performance which is not related to the mapping of data or initializion of weights.

Overall, the previous experimental results support the notion that increasing system dept does generally improve upon classification performance on average, with the biggest improvements usually occuring between system depth increments of 1, 2 and 3 layers. By visualizing the differences between depth increments, a clear advantage shown by increased depth is the ability to produce much more complex mappings of data.

However, in case, increased system depth does not necessarily equate to improved performance. This implies that an optimal depth is data dependent, where depending on the complexity of the task, more layers are needed to effectively seperate each class cluster towards their respective target states. Due to the innate randomness of the weight initialization scheme incorporated, it is hard to justify the impact that this, or other reasons, had on classification performance, i.e., whether any drops in performance were related to suboptimal initial weight choices, slight overfitting to training data or purely from the system depth specified. It could be argued that fixing the initial weightings could provide additional insight into the correlation between depth and classification performance, however this may have negatively affected the upper classification performance benchmark if not chosen suitably.

5.3.3 Introduction Of Noise

Amplitude Damping Noise

To simulate the effect of noise during training in a QPU environment, amplitude damping channels are implemented within the system after each unitary gate operation. Whilst there are many quantum noise channels which could be used to simulate environmental noise (i.e. bit-flips, de-phasing and de-polarizing channels), amplitude damping channels were implemented to provide a realistic noise model, that is also frequently used in the field to model environmental noise [154, 155]. Amplitude damping is a model of qubit energy relaxation through interactions with the environment over time. The result Q, with decay probability $\gamma \in [0, 1]$ of Kraus operators K acting on the density matrix ρ is:

$$Q_\gamma(\rho) \;=\; K_0 \rho K_0^\dagger \;+\; K_1 \rho K_1^\dagger \tag{5.6}$$

Where:

$$K_0 \;=\; \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix} \quad , \quad K_1 \;=\; \begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix} \tag{5.7}$$

The effect that the amplitude damping channel has on the qubits' density matrix can be defined as:

$$Q_\gamma(\rho) \;=\; \begin{pmatrix} \rho_{00} + \gamma\rho_{11} & \sqrt{1-\gamma}\rho_{01} \\ \sqrt{1-\gamma}\rho_{10} & (1-\gamma)\rho_{11} \end{pmatrix} \tag{5.8}$$

For all experiments using simulated noise, amplitude damping is implemented within the mized state simulator available through the PennyLane library. More information on the amplitude damping channel can be found at [30].

Noise Effect Analysis

For the following experiments, artificially-generated datasets of 3 and 15 dimensions were initialized, with a train to test image split of 50 to 150 datapoints per class. For each dataset dimensionality, the noise magnitude was incremented from 0 to 1, in intervals of 0.1. The system depth was also increased in various intervals. Each training instance consisted of 30 epochs of the data, using SGD for optimization and a learning rate f 10^{-2}. In each experiment, final loss values were taken at epoch 30, and the change between these values per noise magnitude λ was subsequently recorded.

As instances of simulated noise was implemented after each parametrized unitary operation, naturally the occurences of noise will increase in proportion to the defined system depth and task dimensionality. Therefore, the recorded change in loss is measured in proportion to the occurences of noise in the experiment setup. This is done to avoid any unfair advantages that a system with lesser depth may possess, since noise would naturally be implemented less than a system with higher depth.

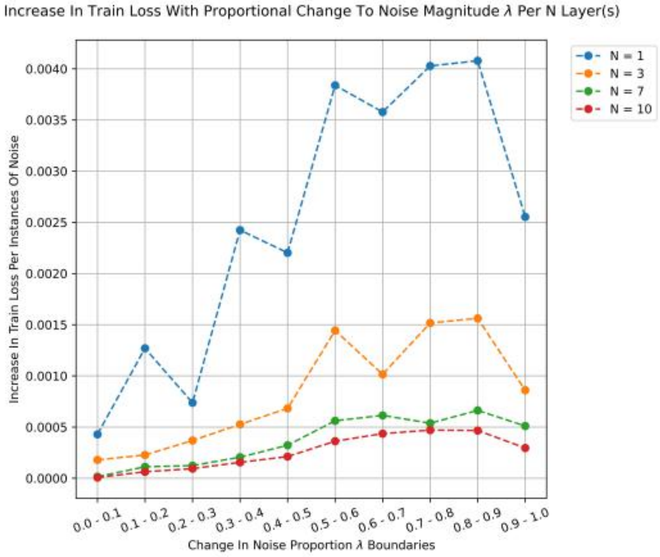


Figure 5.15: Plot displaying the porportional change in final training loss values over various noise strength intervals λ for N layers. The final training loss values are gathered from a binary classification task, using 3-dimensional data.

Within figure 5.15, it can be seen that for the lower system depths of 1 and 3, the increases to the final training loss values are much bigger in comparison to the larger system depths as noise magnitude increases. However, as the number of layers is increased, this rate of change does begin to saturate. In this case, the visible drop-off in training loss to the right-hand side can be justified. For a binary classification task with chosen target states of $|0\rangle$ and $|1\rangle$, as noise magnitude increases, the distribution of datapoints will be drawn closer towards the $|0\rangle$ state. If the noise magnitude is extremely strong, then the datapoints corresponding to target state $|1\rangle$ will be very far away, unable to be drawn further away at all. As the loss value is calculated using the measure of fidelity between states, this explains the corresponding drop off.

Again, as shown in figure 5.16, the rate of increase in loss is much larger for a lower system depth used, even with the higher dataset dimensionality of 15. In this case, a drop can again be seen to the right-hand side of the chart, as a result of the decrease in fidelity between datapoints and their target states slowing down.

The results displayed in figures 5.15 and 5.16 suggest that using a system of larger depth may perhaps bring an advantage of robustness against the influence of noise during optimization. Whilst the benefits of this did appear

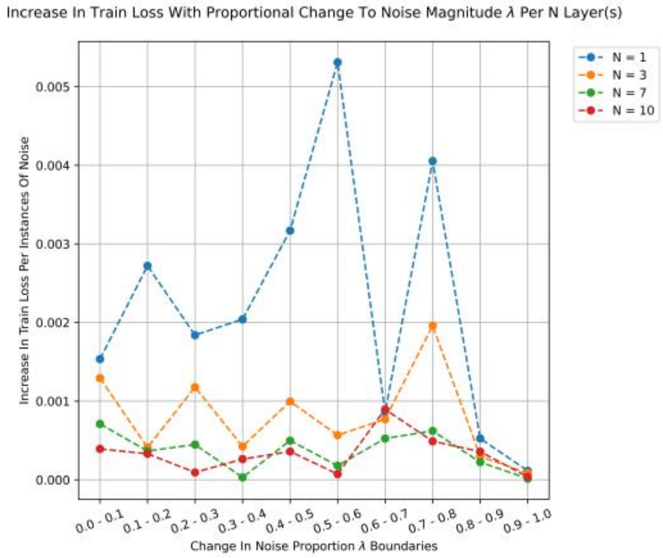


Figure 5.16: Plot displaying the porportional change in final training loss values over various noise strength intervals λ for N layers. The final training loss values are gathered from a binary classification task, using 15-dimensional data.

to saturate as a system depth of 10 was approached, these results do suggest that using additional layers may allow for a better quality of training by resisting the influence of environmental noise, ensuring that the system can converge more stably during optimization.

Figure 5.17 displays the proportional change to training loss for a 3-class classification task using 3-dimensional data. Here, a much sharper classification performance regression can be seen for a single layer as noise magnitude increases. Regardless of the noise increments, 7 and 10 layers show a fairly stable level of increase in loss, only showing signs of divergence from $\lambda = 0.8$ onwards.

In contrast to previous results, there is a diverging behaviour to the right-hand side of the charts. As the results in figure 5.17 were produced from a 3-class classification task, the maximally-distanced target class states are distributed more heavily away from the $|0\rangle$ state in our setup. because datapoints are drawn closer towards the $|0\rangle$ state as noise magnitude is increased, the associated loss value will increase at a higher rate than that of a binary classification task due to a larger cumulative distance between each datapoint and its' target class state.

Similar behaviour can be seen in figure 5.18, showing results from a 3-class classification task using 15-dimensional data. Here, the system depths of 7 and 10 show consistently lower changes in comparison to lower depths of 1 and 3. However, again there is a much larger difference between 1 layer and 3 layers than 7 and 10 layers. This implies that whilst additional layers

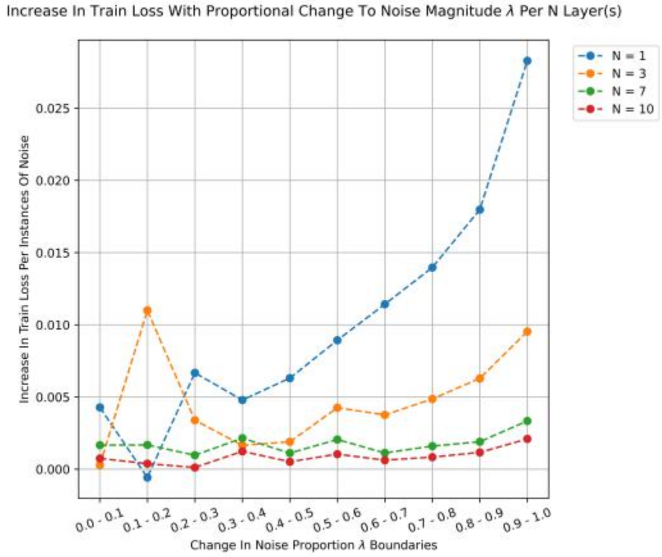


Figure 5.17: Plot displaying the porportional change in final training loss values over various noise strength intervals λ for N layers. The final training loss values are gathered from a 3-class classification task, using 3-dimensional data.

may provide added robustness against the effects of noise during training, this benefit does saturate as subsequently more layers are introduced to the system.

Overall, the results displayed throughout section 5.3.3 support the notion of an advantage of robustness against noise during training, as additional layers are implemeneted to the system. Whilst plots showed a higher variance of training loss between noise magnitude intervals using 15-dimensional data (shown in figures 5.16 and 5.18) over 3-dimensional data (shown in figures 5.15 and 5.17), a higher depth consistently showed more stability throughout experiments, rather than the diverging behaviour seen in lower system depths.

However, similar to the results seen throughout section 5.3.2, any advantage of rubustness during optimization appeared to plateau as depth increased. The sharpest of improvements could be seen between system depths of 1 layer and 3 layers. The differences seen between system depths of 7 layers and 10 layers were minimal, and arguably not worth the increase in complexity and optimization time that the additional layers would bring.

5.3.4 MNIST Case Study Application

Previously, experiments have been conducted using artificially-generated datasets. Whilst this is acceptable for examining specific details surrounding classification performance of an algorithm, it does not always provide a

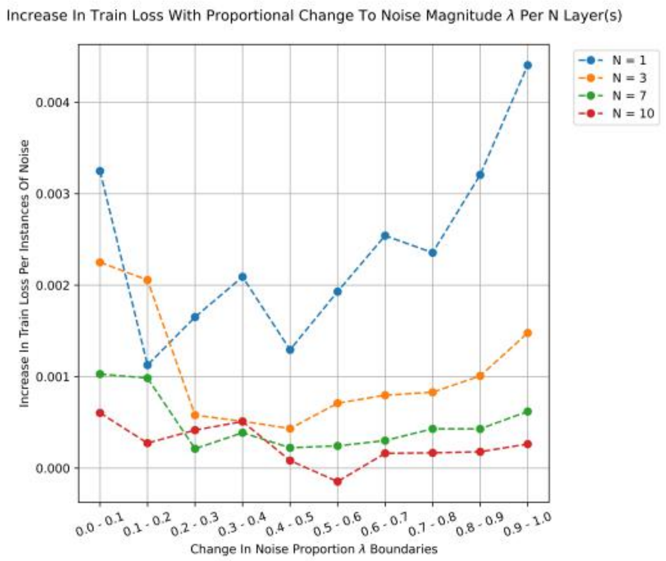


Figure 5.18: Plot displaying the porportional change in final training loss values over various noise strength intervals λ for N layers. The final training loss values are gathered from a 3-class classification task, using 15-dimensional data.

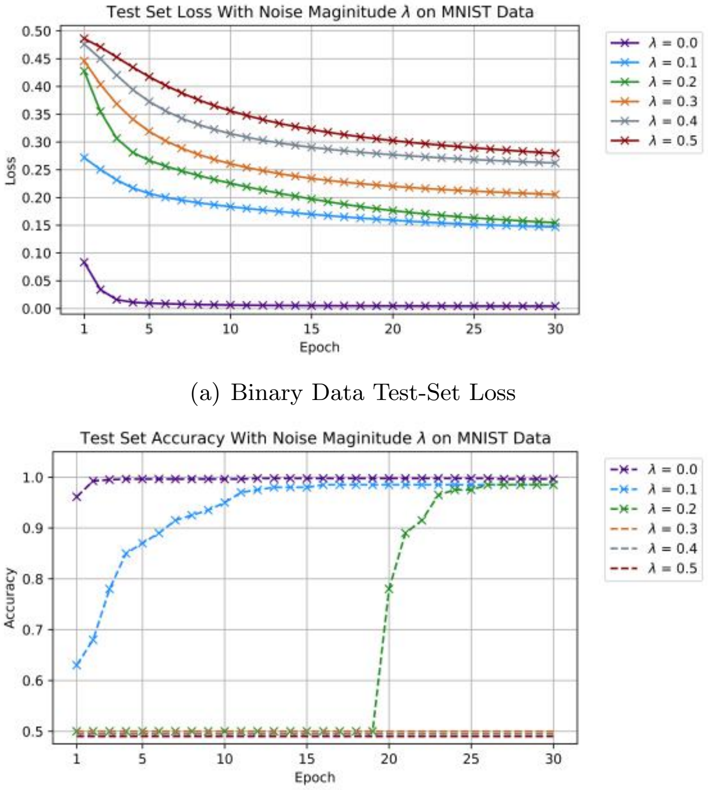
realistic representation of how the algorithm may perform when applied to non-artificial data.

For this reason, the experiments conducted throughout section 5.3.4 will be using a portion of the MNIST dataset. For the experimental setup a subset of MNIST data was used, which consists of 200 images per class for the training portion and 100 images per class for the testing portion. Prior to use, each image is normalized and downsampled to a size of 9x9 using a bilinear interpolation scheme, in order to reduce the overhead processing time required. Whilst doing this means that the results cannot be directly interpreted to unchanged MNIST data, however the change is minor enough that the results still provide a good indication.

For the hyperparameter selection, a system depth of a single layer was used to again reduce the amount of processing required. For optimization, the Adam [156] optimizer was implemented using a learning rate of 10^{-4}. Each individual experiment was optimized over the course of 30 epochs, with test set data results recorded at the end of each training epoch.

In order to give a representation of the impact of environmental noise on non-artificial data, the same scheme for noise introduction outlined in section 5.3.3 was implemented here. For these experiments, noise magnitude values were incremented from 0 to 0.5 in intervals of 0.1, giving a total of 6 experiments per each classification task.

Looking at the results of a binary classification task (using digit classes of 0 and 1) displayed on the left-hand side in figure 5.19, it can be easily



(a) Binary Data Test-Set Loss

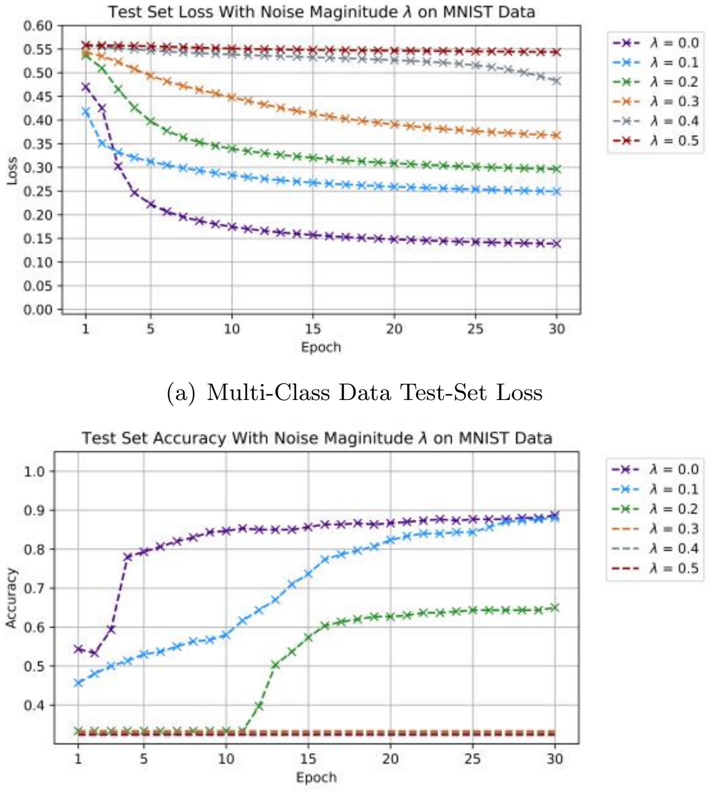
(b) Binary Data Test-Set Accuracy

Figure 5.19: Figure displaying results from a binary (classes 0 and 1) experiments using downsampled MNIST data with varied noise magnitude levels λ and a system depth of 1 layer.

seen that the experiment with zero noise performed to an excellent standard within 5 epochs. As the noise magnitude is increased, the system is still able to classify the test dataset to an excellent standard between noise magnitudes of $\lambda = 0.1$ and $\lambda = 0.2$. However, the loss value begins to converge to 0.15, making this a more realistic level to interpret than the loss result with zero noise influence.

For results with $\lambda = 0.2$, a delay in accuracy increase can be seen, where improvements do not occur until approximately epoch 19. As the loss value is decreasing at a satisfactory rate, this delay can be justified from the distribution of all datapoints residing in a single class state region only until this time. Once a noise magnitude of 0.4 is reached, the system is unable to produce any effective encoding of data and performance is at a minimum throughout.

By looking at the results using 3 classes (digit classes of 0, 1 and 2) displayed on the right-hand side of figure 5.20, the performance is further decreased than when classifying 2 classes only, with a max test-set accuracy



(a) Multi-Class Data Test-Set Loss

(b) Test-Set Accuracy

Figure 5.20: Figure displaying results from a multi-class classification (classes 0, 1 and 2) experiments using downsampled MNIST data with varied noise magnitude levels λ and a system depth of 1 layer.

of 88.7% with zero noise influence. As noise is increased to a value of $\lambda = 0.1$, similar levels of performance are reached, with a final test set accuracy value of 88%, however convergence is notably much slower than when no noise is present.

As noise becomes stronger to $\lambda = 0.2$, classification performance begins to plateau after epoch 15, despite a sharp improvement between the period of epochs 11-15. Similar to binary classification results, once the level of noise reaches a magnitude of $\lambda = 0.3$, the system is unable to perform at a level better than random, since all datapoints will be collected and contained in a single target region.

Overall, the results shown in this section using MNIST data are very promising within the binary classification task, for noise levels up to $\lambda = 0.2$, even with the increase in time for convergence. When extended to a 3-class classification task, classification performance did have a substantial drop, however the target state boundaries are also smaller as additional classes are used.

For both tasks, as noise magnitude levels reach higher values, the system struggles to cope with the influence brought and is unable to converge to a level different than random guesses. These results were with a single uploading layer only, where previous results show adding additional system depth may provide the robustness and complexity of embedding needed to classify the datapoints to a higher standard of accuracy.

5.4 Discussion

The aim of this research is to bridge knowledge between previous works by determining any correlation between system depth and performance using the data re-uploading methodology, as well as measure robustness of the system when using different depth values and finally provide an indication of how the methodology may perform using data that is not of a synthetic nature.

From the experimental results obtained throughout this chapter, a general trend has been identified where increasing circuit depth does tend to improve upon previous classification performance. However, the sharpest improvements to performance seem to occur between 1-3 uploading layers. After approximately 3 layers were introduced, any performance increases were often not as distinctive, where the increased depth generally added complexity to the system with little reward in performance.

In some cases, it can be seen that increasing the system depth did not relate to improved classification performance. For example, this occured in the case of 10 uploading layers trained using 15-dimensional data, which is displayed in figures 5.9 - 5.12. In some of these cases, the behaviour can be justified as a slight overfitting to the training data used, or perhaps from other factors such as the initial weight selection. In these cases, it is hard to determine whether any regression in performance occured solely from the selected system depth, or from the influence of other factors.

A significant advantage that is shown with an increased system depth is the allowance for more complex embeddings of classical data to Hilbert space. Lower circuit depths with a subsequently lower total number of parameters were fairly rigid in their embedding capability, therefore restricting the freedom of movement needed in order to effectively separate the overlapping datapoint clusters.

When examining from a perspective of environmental noise, the results obtained support that a higher system depth can be linked to robustness of noise during training. In comparison to lower system depths, higher depths had consistently smaller proportional changes an the level of noise magnitude increased, therefore providing a more stable platform to optimize from. However, similar to experimental results regarding general classification performance, this advantage of robustness did saturate as additional layers were implemented, with the sharpest robustness improvements generally occuring between 1-3 uploading layers.

In the case of experimental results using MNIST data, the data re-uploading methodology showed promising binary classification results using only a single layer of 162 total parameters. As the levels of noise increased here, a good condition of performance was still achieved with lower noise levels, but was unable to converge after a noise magnitude of $\lambda = 0.3$. Whilst the results gathered using MNIST data may not be state-of-the-art, it should be considered that this performance was achieved using a single layer and a single qubit only. Therefore, these experimental results are fairly promising in relation to the early state that QML is in. As the methodology expands to using multiple qubits with increased depth, then this initial performance can only hope to be improved upon and extended to higher numbers of classes.

Within the wider field of QML, insights gathered from the experimental results are able to be linked to other relevant works directed towards VQC design and implementation. In [142], expressability of a qubit was determined by its ability to navigate the Bloch sphere, which was also analysed in [143]. The experimental results contained in this chapter also support the idea that increased embedding complexity, which relates to the expressability of a qubit, can allow for the necessary complexity of feature spaces required to separate entangled clusters of datapoints. Therefore, by improving the complexity of embedding, or expressability of a VQC, we are able to provide a much higher capability to classify difficult, overlapping datapoint clusters to a good level of performance compared to if this was not considered in our VQC design. However, once a sufficient level of embedding complexity, or expressability has been reached, then implementing additional depth may just increase computational complexity and demand with little improvements to classification performance in return.

From the research presented in this chapter, there are some limitations and areas for future exploration that should be addressed alongside the described contributions. Whilst the experimental results obtained showed trends appearing, it was hard to justify in cases whether performance differences were influenced by other factors such as initial weight distributions or not. Whilst the weight initialisation strategy implemented was kept constant throughout, it was not necessarily an optimal choice. Currently, there have been some efforts to address weight initialisation strategies within a VQC, in particular to avoid barren plateaus [149]. However, it still remains an open question of whether there are any optimal initialisation strategies that may benefit the training of VQCs, and specifically when using the data re-uploading methodology.

When using the data re-uploading methodology with a single qubit only, as we increase the number of the classes used the corresponding class region within the Bloch sphere becomes proportionally reduced in area. In the original proposal of the methodology [125], the authors presented the use of multiple qubits, which naturally introduces a larger state boundary size per class than when using a single qubit only. In order to do this, entangling layers of controlled-Z (CZ) gates were introduced to provide the dependency required between qubit states. However, this also leaves room to investigate the effects from various alternate implementations of entanglement measures, and whether there is an optimal setup for the introduction of multiple qubits.

5.5 Conclusion

In this chapter, a thorough analysis of the data re-uploading methodology was conducted, using the computational power of a single qubit only. Multiple values of circuit depth were used throughout the experiments in order to provide an indication of how this parameter within the system setup affects classification performance.

Here, the experimental results gathered support that increasing depth does improve classification performance of the algorithm, where the sharpest of improvements occur between approximately 1-3 uploading layers used. A clear advantage displayed is that the complexity of data embedding improved alongside increased depth, which allowed for highly overlapping datapoint clusters to separate much more effectively. However, these results also suggest that once a sufficient level of embedding complexity is found, then additional depth may just increase the complexity and computational load with little reward of performance benefit in return.

In the case of experiments simulated within an noisy quantum environment, the results suggest that higher depth values may allow for improved stability during optimization. However, regardless of this extreme levels of noise will continue to have large consequences, due to the nature of the algorithm and how predictions are obtained.

Overall, via the experiments conducted within this chapter, it has been shown that a single qubit only is required to create a sufficiently complex feature space capable of classifying complex data correctly. In comparison to research conducted in chapter 4, the requirement for a single qubit only presents a much larger advantage for coherent results over many qubits, and is therefore a promising method of information encoding and classification to pursue further.

A potentially limiting factor considered with this method is that because singular data dimensionalities are being encoded independent of one another, then this method may perform to a lower standard for classification tasks where data relationships are important, such as image classification. With this said, results of the case study conducted on MNIST data were indeed promising. However, as MNIST data can be classified to a high standard using a classical MLP network which is the most analagous algorithm to data re-uploading, it is hard to justify whether this will translate to tasks that require pixel relationships to be maintained. How pixels are encoded using single-qubit encoding should be investigated and adapted further to maintain these relationships between pixels and progress development towards effective quantum image classification.

Chapter 6

Efficient Quantum Image Classification Via Single Qubit Encoding

Abstract

6.1 Introduction

The following chapter presents the proposition of a quantum image classification methodology that is targeted towards maintaining minimal resources and parameters in order to reduce algorithm complexity. This is conducted following research presented in chapter 5 in an effort to answer the following research question:

Research Question 4: *Can a novel quantum machine learning classifier be developed for image classification that makes efficient use of resources provided?*

The processing capability of readily-available GPU units has enabled a chain of strong-performing DL methodologies [19, 20, 157–162] to dominate the field of image classification, boasting high levels of classification accuracy that can be fine-tuned to a specific task. The result of this is that ML has been able to become integrated within society for many social and industrial uses, e.g., healthcare [163–166], public-safety [167–169] and assisted-living [170–173].

Whilst the current state of DL provides algorithms that can classify complex datasets to a high standard, further improvements are becoming more and more marginal, and are often at the expense of adding many additional parameters. This monumental increase in parameter counts accelerated by GPU capability is not necessarily a negative when the highest-levels of performance are required. However, in order to progress towards effective ML algorithms, the current tradeoffs of requiring additional parameters for marginal gains may not be the most appropriate course of action. The story of deep learning has shown that, by focusing on the development of methods which have a more efficient usage of parameters, a foundation can be provided to build upon and progress towards the highest-performance levels of classification whilst keeping efficiency of training and execution a primary factor.

In chapter 5, experimental results determined that a single qubit alone could classify sufficiently complex datasets to a high degree of accuracy. This makes the method of single-qubit encoding particularly appealing to pursue in an effort to address the challenges described. Therefore, the aim of this chapter is to progress the thorough research conducted towards single-qubit classifiers and propose an architecture that makes efficient use of assigned parameters, as well as improving scalability to higher-dimensional image classification tasks. To do this, the initial experiments contained in this chapter showcase results conducted on lower-level image-classification tasks, following the natural progression of dataset complexity. These experiments are also conducted using noisy and non noisy simulation environments, in order to provide a reasonable expectation of how the method will perform in the current NISQ era.

The findings from experimental results show that as low as 6 parameters is enough to form a suitably complex feature space, capable of classifying image data to a high degree of accuracy. Alongside this, the experimental results show a factor of robustness against the phase damping environmental noise channel to some extent. To summarise, the key contributions of this chapter are two fold: (i) a method is proposed that maintains spatial relationships of pixels through the use of parametrized filters and (ii) the proposed method is adapted to process images in their natural form thus not requiring a costly image flattening pre-processing step.

The structure of the rest of this chapter is as follows. Firstly, related research is discussed, leading to the derivation of the proposed method via single-qubit encoding principles. Then, the setup of the experiments is described in relation to the current capabilities of QML classifiers. Afterwards, the gathered experimental results will be described, where an analysis will be provided. Finally, a discussion of the results and analysis will be conducted in relation to the scope of the field, where potential avenues for future work and extensions to the method may apply.

6.2 Methodology

6.2.1 Related Work

In its current state, NISQ QML algorithms tend to use a backbone of VQCs, where many single-qubit and multi-qubit gates are applied using a set of parameters over a number of qubits [10, 11, 69, 115, 139, 141, 144]. Some of these VQC algorithms are presented as a hybrid approach to computation, working in conjunction with typical classical processes to determine a classification result. Hybrid approaches of computation may provide an opportunity to utilize the power of quantum computing with re-determined methods, e.g., classically extracted features fed through a VQC [36], or vice-versa with quantum-extracted features [88]. However, it may not always be clear to distinguish the impact that quantum processes have on corresponding classification results.

Motivated once more by the success of CNN methods, recent works have proposed fully quantum architectures, mimicking the traditional convolutional-pooling layer series by applying successive unitary operations followed by qubit measurement [12, 82].

Whilst many proposed methods described previously have had promising results and show positivity towards the development of effective quantum classifiers using many qubits, it is important to remain in context with the current NISQ era of quantum computation and understand that minimizing the number of qubits required should be a primary concern when designing quantum algorithms. This is because qubit coherence is not necessarily at the desired standard yet to rely on complex, multi-qubit operations, where a small error could vastly impact the states of other qubits utilized. By focusing development towards smaller scale, efficient methods using minimal qubits, a solid foundation can be built to progress from in the quest for effective QML image classification algorithms.

In an effort to find efficient, yet effective data encoding schemes, recent works [142, 143, 154] have analyzed a variety of QVC structures to determine the ability of the encoding to navigate the Bloch sphere, capability of entanglement between qubits, as well as robustness when realized in a noisy quantum environment. A particular method has achieved promising classification results whilst only requiring the use of a single qubit to function [125]. This was furthered in [174], where the single-qubit classifier was still found to remain effective for a multitude of tasks, even in noisy quantum environments.

In summary, a qubit is an extremely powerful computational tool, such that the development towards quantum classification methods should have a primary focus to maximize the usage of each qubit prior to increasing the amount of them. By doing so, a solid foundation can be built to progress forwards from in an effort to create effective, robust quantum classifiers, similarly to the rise of state-of-the-art DL methodologies dominating many classical ML problems.

6.2.2 Single-Qubit Encoding

To preface the description of the proposed methodology, it is relevant to discuss a particular method of quantum information encoding, known as single-qubit encoding. For many ML tasks, data is often presented in the form of a column vector. Traditionally, this D-dimensional vector of classical data could be encoded by initializing a 2^D qubit quantum state as a binary string equivalent (basis encoding) if applicable, or through translating data dimensions into their corresponding probability amplitudes of a superposition state (amplitude encoding). Whilst these data encoding schemes have been employed within other works [47, 57, 175], they are often very costly or impractical to implement, and can become susceptible to error-prone quantum operations. Therefore, these encoding schemes may not always be an efficient means of minimizing usage of qubits.

Single-qubit encoding, developed in [125], is a strategy of encoding a vector of classical data into a feature Hilbert space using a succession of unitary operations acting on each input data dimension applied on a single qubit only. For any arbitrary special unitary group of degree 2 SU(2) matrix operation U (a 2x2 unitary matrix of determinant 1), the corresponding operation is able to be decomposed into the following three rotational operations [30]:

$$U = e^{i\alpha} R_Z(\beta) R_Y(\gamma) R_Z(\delta) \tag{6.1}$$

With a global phase factor α, Euler angles $\beta, \gamma, \delta \in \mathbb{R}$ that define the extent of rotation (R) around the Z, Y and Z axes respectively. Within this method of encoding, the Euler angles are parameterized further and defined as:

$$\begin{aligned} \beta &= \theta_i + x_i \cdot \phi_i \\ \gamma &= \theta_{i+1} + x_{i+1} \cdot \phi_{i+1} \\ \delta &= \theta_{i+2} + x_{i+2} \cdot \phi_{i+2} \end{aligned} \tag{6.2}$$

Where θ and ϕ are trainable weight parameters assigned to x_i, the value of input vector x at dimension i. Therefore, the extent of rotation β, γ and δ is with respect to the weighted value of the input. This is defined in the same manner as in section 5.2 Using the previous parameter definitions, a maximum of three input dimensions can be encoded per unitary operation applied.

From here, the input vector will be continually cycled through, encoding a series of three-dimensional values at a time, until the entirety of the input has been encoded. This is known as a full 'upload layer' of the input data.

As an example, for an input vector of 144 dimensions, each dimension will have an associated θ and ϕ variable. Therefore, for this example a total of 288 parameters are required to encode the information fully.

6.2.3 Proposed Methodology

With analogies to classical feed-forward neural networks, single-qubit encoding is an effective way of creating a highly complex feature space through repeated upload layers of input data. However, as information is encoded at a singular-pixel level, it may be at a disadvantage for tasks where it is important to utilize spatial information of pixels, such as image classification. This step of incorporating local regions of pixels is a fundamental aspect of convolutional layers used within DL, where the typical approach is to use a filter, or 'sliding window', that gathers a square region of FxF pixels. In classical ML, a kernel operation would be applied to result in a value for that region of pixels.

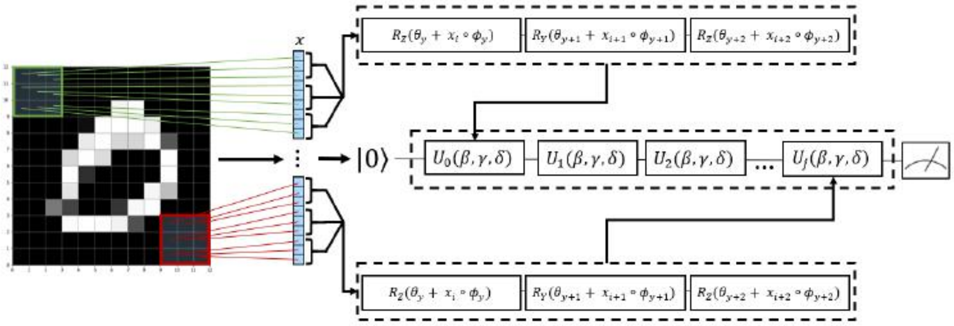

x
$R_Z(\theta_y + x_i \circ \phi_y)$
$R_Y(\theta_{y+1} + x_{i+1} \circ \phi_{y+1})$
$R_Z(\theta_{y+2} + x_{i+2} \circ \phi_{y+2})$
$|0\rangle$
$U_0(\beta,\gamma,\delta)$
$U_1(\beta,\gamma,\delta)$
$U_2(\beta,\gamma,\delta)$
...
$U_j(\beta,\gamma,\delta)$
$R_Z(\theta_y + x_i \circ \phi_y)$
$R_Y(\theta_{y+1} + x_{i+1} \circ \phi_{y+1})$
$R_Z(\theta_{y+2} + x_{i+2} \circ \phi_{y+2})$

Figure 6.1: An overview of the proposed methodology, which shows the filter-pass followed by encoding and measurement.

The first step in the proposed methodoloy is to adopt a similar approach to this. Rather than flatten an image into the form of a column vector as a pre-processing step, the original shape of the image is maintained. A filter of size FxF is then passed over the image, partitioning the image into a distinct grid of FxF squares. Each square region of pixels is then flattened in turn and encoded onto the qubit using the described single-qubit encoding scheme with pixel values (x_i), as well as respective filter weights as parameters (θ_i, ϕ_i). This process is repeated until the filter has passed over the entire image, where measurement is conducted with respect to a target state provided. The number of square pixel regions to encode and the number of unitary operations required j is determined by the size of the filter FxF, as well as the stride value S used.

By adopting this approach, pixel information can be encoded in such a manner where spatial relationships between pixels are maintained. To clarify, rather than assigning a set of trainable parameters to each square

FxF region of pixels, a set of 6 weight parameters are assigned to the filter itself, which correspond to θ and ϕ in equation 2. By doing so, the same set of 6 parameters will repeatedly be applied to every series of 3 pixels that the filter has extracted. This method reduces the number of parameters required to just 6 per filter.

Whilst it is acknowledged that multiple unique sets of 6 parameters could be localized to each FxF region, the aim of this research is to demonstrate that it is possible to produce reasonable results with the fewest parameters, therefore all experiments contained within this chapter will be conducted using a system setup of a single filter with 6 parameters in total, as displayed in figure 6.1. However, both setups discussed offer a slightly different approach towards image classification with advantages and disadvantages for each. This may open various avenues for future work to explore, hence why it is included in this section.

6.2.4 Classification Pipeline And Loss Calculation

So far, the proposed encoding strategy has been defined in section 6.2.3, however the flow from input to classification output has not been made evident. To do this, a fidelity-based approach of measurement is adopted as seen in [125], where the overall objective is to minimize the fidelity between a set of data encodings and their respective target states. For a binary classification task, given a set size D of images with corresponding class values in $0, 1$, a respective target state of $|0\rangle$ or $|1\rangle$ is assigned to each image. Any number of classes can be incorporated using this approach, providing that the target states are maximally-distanced from each other.

From here, the proposed encoding strategy is adopted until all pixel values have been encoded onto the qubit. Once at this point, measurement occurs where the fidelity of the qubit is extracted against each target class state in turn. In short, fidelity F is a measurement of similarity, or closeness between two quantum states, where $0 \leq F \leq 1$. The higher the fiedlity of two quantum states, the more similar they are in direction. The highest class fidelity value given is then considered to be the result of classification. The following loss function is then applied, which is based on that utilized in [125]:

$$\frac{1}{2D} \sum_{m=1}^{D} \sum_{c=1}^{C} \left(\left(F\left(x_d, \theta, \phi\right) - F_c \right)^2 \right) \tag{6.3}$$

Where D is the set size of images used, C is the number of classes, $F(x, \theta, \phi)_d$ is the measured fidelity of the current datapoint (image within the dataset) d with respect to class c and F_c is the expected fidelity value to be measured. To clarify, a datapoint of class 0 has a target state of $|0\rangle$, with expected fidelity values of 1 and 0 for class values 0 and 1, respectively.

If the qubit was in state $|0\rangle$, then the fidelity measurement would equal 1. If the qubit was in state $|1\rangle$, then the fidelity measurement would equal 0. Say the qubit was in a state of $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, then the fidelity measurement is given by:

$$F(x_d, \theta, \phi)_c = |\langle\psi_c|\psi(x_d, \theta, \phi)\rangle|^2 \tag{6.4}$$

Which for $c = 0$, $F(x_d, \theta, \phi)_c = 0.5$. Expected fidelity values can also be found using equation 6.4 by cycling through each class value with one another.

To display the classification process in full, algorithm 1 showcases the classification process from input to output. In short, for each image, filters are passed over extracting square regions of pixels at a time. Following this, unitary operations are performed to the qubit in turn using pixel values from each region with filter weights as parameters. This is repeated until all pixels have been encoded, where fidelity measurements are taken with respect to the class states. To ensure clarity for the statements in lines 12 and 19, the value in line 12 relates to the three required values per unitary gate β, γ and δ, where if x is not a multiple of 3, then a placeholder value of 0 is applied, which has no additional effect on qubit rotation. Line 19 refers to the successive application of unitary operations, where the cycling of i in multiples of 3 allow for the 3 unitary operation values β, γ and δ to be given in turn.

Algorithm 1: Proposed Methodology Process

```
Input: Dataset $D$, Filters $N_f$, Class States $C$, Parameters
       $\beta, \gamma, \delta \in N(\mu = 0, \sigma^2 = 0.1)$
, Array of pixel values $x$ Output: Qubit fidelities $Q$
for image $d \in D$: do                                   // Image of height $H$ and width $W$
    for filter $f \in N_f$ do                             // Filter of size $FxF$ and stride $S$
        while pixel row $d_{row} < d_{height}$ $H$ do     // Cycle Filter Regions
            while pixel column $d_{col} < d_{width}$ $W$ do
                for filter row $f_{row} = 1, ..., F$ do   // Group Pixel Regions
                    for filter column $f_{col} = 1, ..., F$ do
                        $r = (d_{row} + f_{row})$ ; $c = (d_{col} + f_{col})$
                        if $r < H$ and $c < W$ then
                            $x$ append value at pixel $(r, c) \in d$
                        else
                            $x$ append 0
                if $len(x)$ % 3 $! = 0$ then              // % = modulo operation
                    $x$ append 0
                $d_{col} \rightarrow d_{col} + S$
            $d_{row} \rightarrow d_{row} + S$

    for filter $f \in N_f$ do                             // Quantum Circuit Application
        for $\psi_c \in C$ do                             // Cycle Class States
            for $i = 1 : 3 : x_{max}$ do                  // Apply weights in sets of 3
                $\beta = \theta_1 + x_i \cdot \phi_1$
                $\gamma = \theta_2 + x_{i+1} \cdot \phi_2$
                $\delta = \theta_3 + x_{i+2} \cdot \phi_3$
                Apply $U(\beta, \gamma, \delta)$
            Measure Qubit Fidelity $F(x_d, \theta, \phi)_c$ w.r.t $\psi_c$
            $Q_d$ append $F(x_d, \theta, \phi)_c$
```

6.3 Results

6.3.1 Experimental Setup And Overview

In this section, the experiments conducted using the methodology described throughout section 6.2 will be presented, where an initial analysis will be conducted into the results obtained. The first set of experiments will be conducted using a subset of the MNIST dataset, for both binary and 3-class classification tasks. The MNIST dataset [176] is often considered an initial benchmark for many ML systems targeting image classification as their primary task. Due to the early nature of QML algorithms, we feel that using the MNIST data provides a suitable challenge to showcase the lowest performance boundary of the system using minimal parameters.

Following this, as a step up in difficulty our experiments will be conducted using a subset of the FMNIST dataset [177] for binary and 3-class classification tasks. FMNIST data is often considered a subsequently more challenging task following the MNIST data, so will pose an appropriate challenge for the low-parameter system to tackle effectively. Thirdly, the methodology will be applied to a face identification task, using a custom dataset consisting of AT&T face images [178], as well as a collection of images taken at random from the CIFAR10 dataset [89].

Finally, it is important to consider the impact that environmental noise has on the capability of the algorithm presented. Given that the following experiments are conducted in a simulation environment, the noise implementation will also be simulated, but makes appropriate use of various noise and distortion channels to produce realistic results to interprit.

To provide general details of the experimental setup and implementation, the framework for these experiments was developed using the PennyLane library [135], which also incorporated usage of the PyTorch interface [179]. For non-noisy simulations, the Qulacs [152] qubit simulator was used as a plugin to PennyLane. For simulations that introduce noise, the PennyLane native mixed-state simulator was used. For reproducibility, all relevant pseudo-random number generation seeds were set to zero.

For initialization, all weights were formed using a Gaussian distribution with a mean of 0, and standard deviation of 0.1. As a side-note in reference to the general barren-plateau problem largely present in training QVC and similar quantum algorithms [146], it is relevant to address the consideration taken towards this. Whilst it is acknowledged that there have been some proposals towards overcoming the problem of barren-plateaus, namely through localized cost-functions [147] usage of quantum natural gradients [150], and evaluations of initial weight selections [149], no specific approaches to reduce their occurrence were incorporated. Optimization of experiments was conducted as normal, where if a barren plateau was seen to be present, then training would be reconducted using a new distribution of weights. This is not necessarily an optimal method to remove the problem of barren plateaus, however there is no common practice known as of yet to overcome this problem currently, therefore it was felt that this course of action was appropriate for now.

6.3.2 MNIST Dataset Results

For the following results, a subset of the MNIST dataset was used. This subset consisted of 500 training images per class used and 250 test images

Table 6.1: Final classification performance values at epoch 30 for various filter sizes using binary MNIST data.

Filter Size	Train Loss	Train Accuracy	Test Loss	Test Accuracy
3x3	0.06280	0.932	0.06090	0.946
4x4	0.05012	0.951	0.04739	0.958
5x5	0.07104	0.920	0.08148	0.912

per class used. For each experiment, 30 epochs of optimization were conducted using the Adam [156] optimizer with a learning rate of 10^{-4} . These hyperparameter values were selected from a small group of initial experiments conducted in order to find a suitable choice of learning rate for the number of epochs used. 30 epochs of optimization was also selected from initial experimets as satisfactory convergence could be reached within the timeframe, whilst not requiring extremely long training periods.

The results displayed in table 6.1 show classification performance values from experiments conducted using binary MNIST data of classes 0 and 1, with a varied filter size. Here, the train set and test set accuracy achieved was 0.951 and 0.958 respectively for a filter size of 4x4. The second-best performing filter size was 3x3, followed by 5x5 in third.

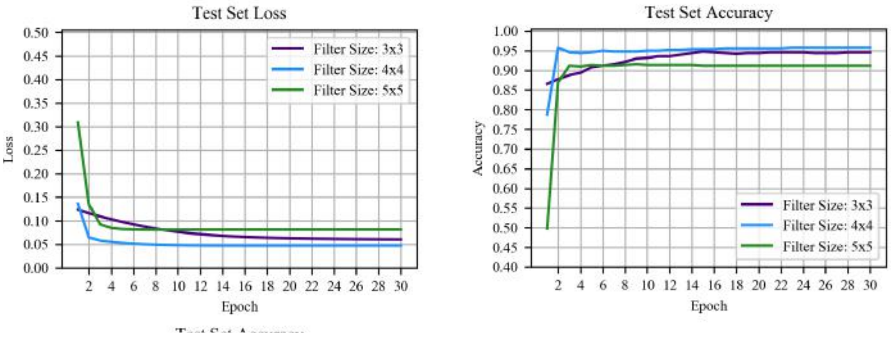


Figure 6.2: Test set loss and accuracy results of a binary classification task, conducted with a single filter of varied size on a subset of MNIST data with classes 0 and 1.

Upon inspection of the test set loss and accuracy curves displayed in figure 6.2, it can be seen the behaviour of the curve for the 3x3 filter is different to that of 4x4 and 5x5. Here, the curve for the 3x3 filter experiment begins at a more favorablestandard of classification performance, but any improvements occur slowly and gradually over the course of training. The latter experiments of filter sizes 4x4 and 5x5 begin in a more unfavorable position with lower loss and accuracy values, however the initial improve-

Table 6.2: Final classification performance values at epoch 30 for various filter sizes using 3-class MNIST data.

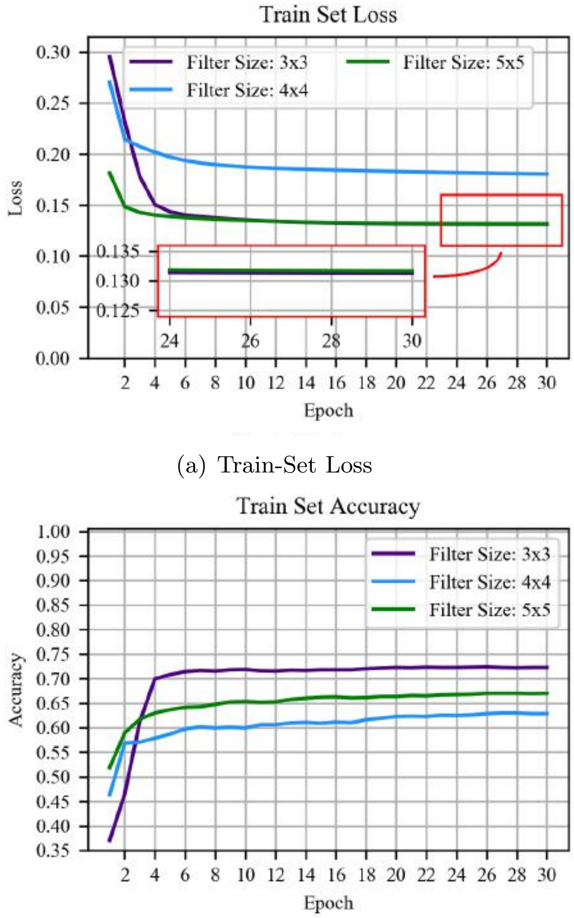
Filter Size	Train Loss	Train Accuracy	Test Loss	Test Accuracy
3x3	0.13127	0.723	0.13337	0.721
4x4	0.18036	0.629	0.17281	0.627
5x5	0.13160	0.670	0.13181	0.659

ments to classification performance are very sharp and quickly plateau by approximately epoch 5. The behaviour exhibited here suggests that whilst the initial weight distribution for the 3x3 filter experiment may classify the dataset to a higher standard to begin with, the starting weight distribution may also be present in a region of lower gradient within the loss landscape.

The slower yet fairly consistent optimization curve supports this, as the system could be steadily attempting to maneuver out of this lower-gradient region. It is unclear whether, given enough training epochs, the experiment using a filter size of 3x3 will overtake the 4x4 filter experiment. However, the 3x3 filter curve does appear to plateau at approximately 22 epochs, therefore this would suggest that the system had settled into a local minimum, and is unable to improve further. Regardless of considerations towards optimal and suboptimal weight distributions and barren plateau regions within the loss landscape, the system is still able to consistently classify the testing portion of the dataset to a high degree of accuracy in the 90% bracket within 5 epochs.

Table 6.2 shows final performance values taken at epoch 30 from experiments conducted on multiclass (3-class) MNIST data using classes 0, 1 and 2 and a varied filter size. Within this, a filter size of 3x3 produced the best classification performance overall, followed by the 5x5 filter and 4x4 filter respectively. Whilst the results achieved here may not be state-of-the-art, there are some points which must be considered in context of this research. The first is that classification is being conducted using fidelity measurements of a set of maximally-spaced target state vectors. As only a single qubit is being examined, the distance between class states becomes smaller as more classes are considered. As the loss function implemented aims to minimize the distance between embedded datapoints to their target class state, this naturally becomes more difficult to achieve with an increased number of classes, providing the dataset is not easily separable.

If the dataset is not easily separable, then the low parameter count implemented here may not be able to provide an embedding capability that is complex enough to account for this. As charts displaying train set loss and accuracy in figure 6.3 show, this lower embedding complexity thus equates



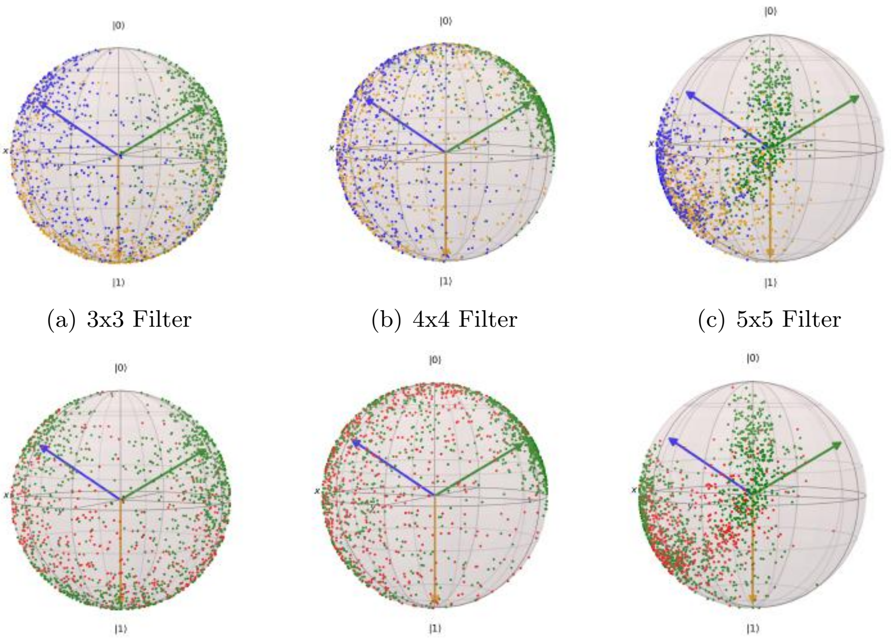
(a) Train-Set Loss

(b) Train-Set Accuracy

Figure 6.3: Train-set loss and accuracy results of the experiment using 3-class (classes 0, 1 and 2) MNIST data with varied filter sizes.

to a plateau, or extremely marginal improvements in both loss and accuracy over time. For additional clarity, the inset box for the train-set loss chart in figure 6.3 displays the 3x3 filter line just below the 5x5 filter line.

In order to demonstrate this, figure 6.4 displays embeddings of train set data during epoch 30 from each experiment as datapoints on the Bloch sphere. This is done to assist in our understanding of how the embedding capability of the current system setup, combined with reduced class area from adding classes affects classification performance. Here, the clearest difference between embeddings is that the 5x5 filter (figures 6.4(c) and 6.4(f))produced a much denser embedding of all datapoints in this case. In contrast, embeddings from a 3x3 (figures 6.4(a) and 6.4(d)) and 4x4 filter (figures 6.4(b) and 6.4(e)) were fairly similar in that the datapoints are



(a) 3x3 Filter (b) 4x4 Filter (c) 5x5 Filter

(d) 3x3 Filter (e) 4x4 Filter (f) 5x5 Filter

Figure 6.4: Visualization of Bloch sphere embeddings at epoch 30 for 3-class MNIST training-set images. Point colours for (a) - (c) correspond to the images respective class, whereas for (d) - (f), green and red points represent correctly and incorrectly classified datapoints respectively. For all Bloch spheres, the three central arrows represent the target state vector for that class.

more widely distributed towards their respective target states overall, with the 3x3 filter experiment arguably showing the most distinctive distributions of datapoints per class. However, despite these differences, the loss value of the 3x3 experiment is very slightly below the 5x5 filter experiment. Yet, when accuracy is considered, this 0.00033 difference in loss equates to over 5% drop in accuracy.

This can be justified by looking at the position of the colour groups of datapoints for the 5x5 experiment in figure 6.4(c). Looking at which classifications are correct (green points on the bottom row), it can be seen that the majority of these correspond to the distinct clusters of blue and green datapoint groups in the plot above (equating to various image classes). However, there is a large section towards the bottom left where there is a significant overlap between the blue and yellow class clusters. This shows that the embedding capability here was not strong enough to separate these clusters as effectively as the 3x3 filter experiment, where the datapoint clusters were spread more widely yet remained fairly compact.

Whilst the 4x4 filter experiment produced the poorest classification performance results overall, the resulting embeddings show that this experi-

Table 6.3: Final classification performance values at epoch 30 for various filter sizes using binary FMNIST data.

Filter Size	Train Loss	Train Accuracy	Test Loss	Test Accuracy
3x3	0.11616	0.884	0.10844	0.895
4x4	0.12951	0.824	0.16690	0.790
5x5	0.17981	0.726	0.25194	0.575

ment struggled to form a significant class cluster consisting of the yellow datapoints, and so had many incorrectly classified images as a result, as depicted in figure 6.4(b). Had the 4x4 filter experiment been more successful in doing this, then it could be argued that the final embeddings of the image data would act similarly to the 3x3 experiment, thus producing a stronger classification accuracy.

Overall, the multiclass MNIST experimental results show that the system is capable of classifying the majority of datapoints in their correct classes with just 6 parameters. However, perhaps this classification and embedding capability could be improved by further experiments and analysis into the system design, i.e., including additional filters.

6.3.3 FMNIST Dataset Results

For the following results, a subset of the FMNIST dataset was used. This subset consisted of 250 training images per class used and 100 test images per class used, both sampled at random from the entire dataset. For each experiment, 30 epochs of optimization were conducted using the Adam optimizer with a learning rate of 10^{-3}. These hyperparameter values were selected as a result of conducting a small group of initial experiments to find a suitable choice of learning rate for the number of epochs used.

Table 6.3 displays classification performance values from experiments conducted using varied filter sizes. From these results, a filter size of 3x3 was the best performing filter size, reaching a test set accuracy close to 90%. Unlike results using binary MNIST data, classification accuracy regresses as the filter size is increased.

When inspecting the train set and test set loss curves displayed in figure 6.5, the behaviour of all three experiments appears to contradict one another to some extent. Whilst the 3x3 filter size experiment initially performs worse than the others, it shows a very rapid decrease in loss, followed by a sharp plateau. In contrast to this, the 4x4 experiment shows a slow and gradual decrease in loss, and the 5x5 experiment displays very little convergence and plateaus close to the initial loss value after epoch 1. The

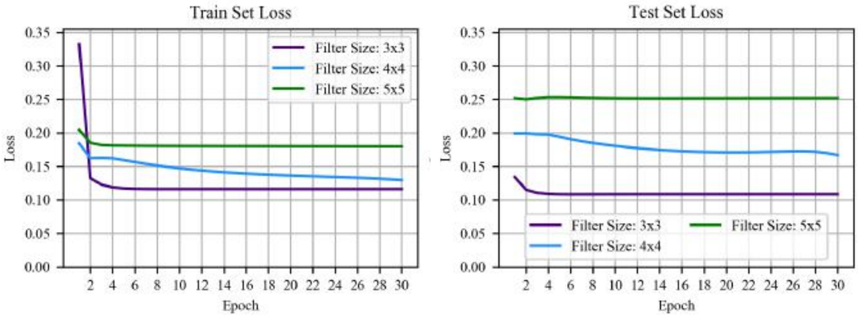


Figure 6.5: Train set and test set loss curves relating to experimental results displayed in table 3. The experimental data consisted of a subset of the FMNIST dataset using image classes of 0 and 1.

charts displayed in figure 6.5 appear to suggest that the embedding capacity of the algorithm in its current state is perhaps not complex enough to be able to optimize effectivelyto the training data provided. The difference in loss values at epoch 1 is likely caused from the initial weight distributions for each experiment being in more advantageous starting positions.

The sharp decrease in loss that follows for the first 2-3 epochs could then be explained by the system attempting to separate the cluster of datapoints formed at the start to their respective target states. However, the complexity of embedding that a single filter provides is perhaps not too great, meaning that the datapoints which are of a fairly similar nature are unable to be separated further into two opposing class clusters. This results in the overall distribution of datapoints on the Bloch sphere being left virtually unchanged, hence a plateau in the loss value itself. In addition, the 4x4 filter experiment could be in an area of lower gradient within the loss landscape, resulting in the behaviour displayed and described earlier being drawn out over a longer period of time.

In order to visualize this, figure 6.6 displays various Bloch sphere embeddings of train set data at epoch 1, 2 and 30 with point colour corresponding to class value. For the left-hand Bloch sphere plot at epoch 1, the distribution of datapoints is fairly dense towards to top hemisphere close to state $|0\rangle$, the target class state for class 0. As the loss function implemented refers to the fidelity, or measure of distance between the datapoints and their respective target classes, the fact that many datapoints of class 1 are far away from their target state of $|1\rangle$ will cause the loss value to increase.

Following a single epoch of training, the second set of embeddings for epoch 2 are more evenly distributed between the two hemispheres. Visually, as the datapoints are embedded closer towards their target state on average, this is equated to the prior considerable drop in loss value. However, between epoch 2 and 30 the system is unable to separate the two clusters of datapoints

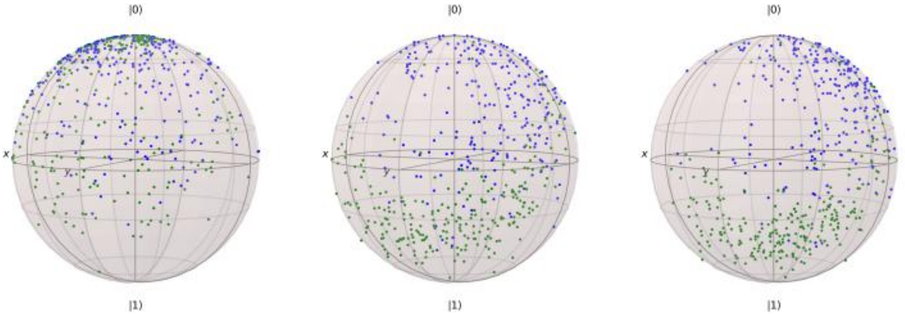


Figure 6.6: Visualizations of Bloch sphere embeddings of datapoints corresponding to dataset images for the experiment conducted using a 3x3 filter size. Left to right Bloch spheres show train set data embeddings taken over epoch 1, epoch 2 and epoch 30. Point colours correspond to the images respective class, where blue points represent class 0, and green points represent class 1.

and embed them closer towards their respective target states. In particular, there is an area along the right-hand side of the Bloch sphere that contains an overlap of the two class clusters of datapoints.

Because the system is unable to separate the datapoints located within this area, the overall shape of embeddings is simply shifted around equally, meaning any decrease in loss for a particular class is mirrored by an increase for the opposing class. This causes the overall loss value to be left fairly unchanged, hence the plateau described earlier. If the complexity of embedding was higher, then perhaps the system could separate the class clusters of datapoints much more effectively, resulting in a continued convergence of loss towards a lower value and a higher accuracy in time.

With these points considered, even with the suggested lowest level of embedding complexity that the system offered within this experiment, a test set classification accuracy close to 90% was achieved. In the context of this research, this is a promising achievement, which can only hope to be improved upon if the embedding capability, followed by the subsequent learning capacity of the algorithm was increased through additional research and analysis.

6.3.4 Facial Identification And Facial Recognition Results

The following subsection consists of two experimental setups. The first set of results consists of a bespoke dataset that was created and consists of images from the AT&T face dataset with a random selection of images taken from the CIFAR10 dataset. The objective of this experiment is to determine whether a provided image is that of a face (class 0), or non-face (class 1). A training set of 300 images per class was used, and a testing set of 100 images per class was used. For each experiment, 30 epochs of optimization were conducted, using the Adam optimizer with a learning rate of 10^{-3}. As

Table 6.4: Final classification performance values at epoch 30 for various filter sizes using custom facial identification data.

Filter Size	Train Loss	Train Accuracy	Test Loss	Test Accuracy
3x3	0.14685	0.808	0.15141	0.825
4x4	0.17433	0.763	0.17231	0.745
5x5	0.17580	0.752	0.16917	0.780

with all experiments, all hyperparameter values were selected by conducting a small group of initial experiments in order to find a suitable choice of learning rate for the number of epochs used. As before, 30 optimization epochs allowed for satisfactory convergence without excessively lengthy training periods.

Experimental results with loss and accuracy as classification performance metrics can be seen in table 6.4. As with previous multiclass MNIST and FMNIST experimental results, the filter size of 3x3 produces the highest performance overall.

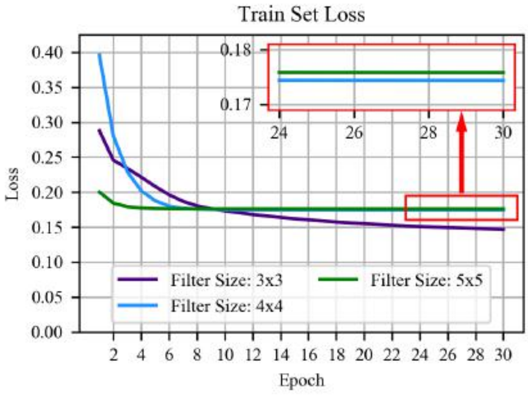


Figure 6.7: Train set loss results of a facial identification task, using a single filter with varied size between experiments. The dataset consisted of a AT&T face image data, combined with a selection of images taken from the CIFAR10 dataset.

When viewing the graph of training set loss, displayed in figure 6.7, the loss values for 4x4 and 5x5 filter size experiments are very similar, and appear to plateau at the same epoch. However, the loss curve for the 3x3 filter experiment does not appear to plateau in this experiment over the number of optimization epochs conducted.

By visualizing the associated image embeddings figure 6.8, it can be seen that at epoch 10, the two class distributions are heavily overlapped at the border between the two classification regions (the two hemispheres in the

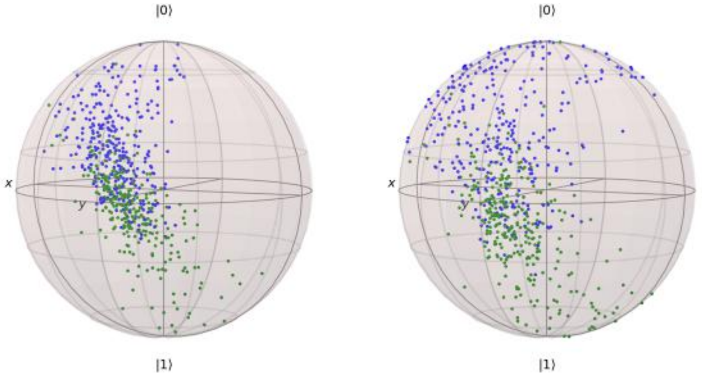


Figure 6.8: Bloch sphere visualizations of train set image embeddings for the 3x3 filter experiment in figure 6.7. The left-hand sphere shows embeddings during epoch 10, whereas the right-hand sphere shows embeddings during epoch 30. Point colours correspond to the class of the embedded image, with blue for class 0 and green for class 1.

case of binary classification). As optimization continues by epoch 30 it can be seen by the right-hand Bloch sphere that whilst the datapoint clusters are still overlapping around the central axis, they are being drawn away from each other slowly.

This equates to the slow but gradual decrease in loss seen throughout training for the 3x3 filter experiment, where datapoints are becoming closer to their respective target states, just at a slow pace. This behaviour suggests that because many datapoints are located close to the boundary between the two class regions, even a small separation between the two interlinked clusters could produce a relatively large increase in accuracy. However, it is also unclear where the natural limit of the system is in this case, and a plateau could be reached at any moment.

Regardless of any speculative analysis, the results achieved here are once again promising, and support the aims of this research by showing that a good classification result can be achieved with few parameters needed, providing a foundational algorithm with potential for further development and improvement.

The following results are from the second experimental setup within this subsection. The objective of this experiment was to perform a facial recognition task, using different individuals from the AT&T dataset. Due to the small size of individual class subsets within the dataset, it felt appropriate to include these results as an additional small-scale experiment, following the previous facial identification experimental results that used a larger scale of data. Here, a training set of 7 images per class and a testing set of 3 images per class was used, with two classes of image in total.

For the results displayed in table 6.9, the classification accuracy for the

Table 6.5: Final classification performance values at epoch 30 for various filter sizes using binary AT&T facial image data.

Filter Size	Train Loss	Train Accuracy	Test Loss	Test Accuracy
3x3	0.23605	0.643	0.11310	0.833
4x4	0.23500	0.571	0.24201	0.500
5x5	0.23895	0.429	0.33591	0.500

training set of data was fairly poor for all experiments. Whilst the testing set accuracy was fair for the experiment using a 3x3 filter size, the other experiments produced an even guess for each class. The unusual set of results achieved here could suggest that there was simply a too small scale of data to truly learn an existing representation between the opposing classes.

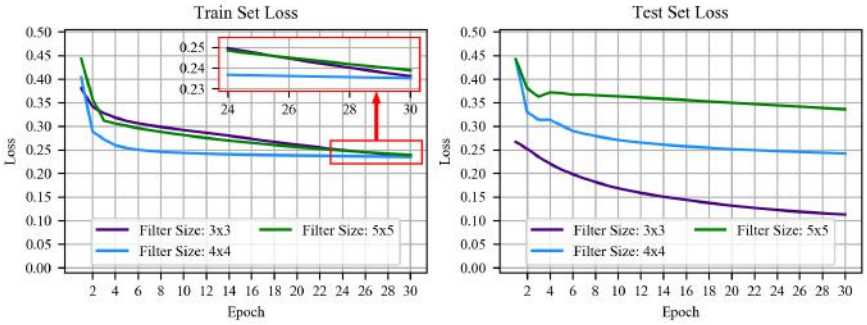


Figure 6.9: Loss result curves for a binary face recognition task conducted using AT&T face image data.

This is again supported by result graphs shown in figure 6.9, as the training loss for each experiment appears to plateau at very similar values, determining that the system had perhaps reached its' natural limit with the data provided. In contrast to this, interestingly the curve for testing set loss continues to decrease regardless of the previously mentioned plateau. This could be explained by the initial weight distributions affecting the end embedding result for the test set data. In other words, the graph would suggest that the experiment using a 3x3 filter size was initialized with a more optimal selection of weight values than the others, therefore allowing the subsequent embeddings of test data datapoints to be on average more in their respective class regions.

Another point that should be considered is the nature of the task itself. Whilst the aspect of small-scale data has been mentioned, an important step within many facial recognition methods is the feature extraction step. This step allows algorithms to extrapolate key characteristics of an individuals face to aid in classification. As a feature extraction step was not introduced within this methodology, then combined with the small amounts of data provided, the system struggled to learn any representation and difference between the two individuals. Better results may have been achieved if a feature extraction pre-processing step was introduced, however this is beyond the scope of this research and is a topic to be explored if the algorithm was specifically applied to a facial recognition task.

6.3.5 Environmental Noise Impact

In the current NISQ era of quantum computing, it is important to consider the effect that environmental noise has during optimization of quantum algorithms. There are two approaches to analyzing the effect of environmental noise. The first is running the algorithm directly through a quantum processing unit (QPU), and the second is by recreating environmental noise using a noisy qubit simulator. Both approaches have advantages and disadvantages to them, but either provide a reasonable insight into how the algorithm may perform in the NISQ era. Due to the ability to monitor the effect of noise more closely, our implementation was conducted using the second approach by simulating environmental noise.

In order to recreate instances of environmental noise, there are various noise channels which can be applied to simulate different effects of noise occurring on quantum information. As an example, various noise simulation channels available include de-phasing, bit-flip and amplitude damping channels to name a few. For the purposes of this section, the environmental noise channels that will be implemented are amplitude damping and phase damping. These models of noise were chosen as they are realistic models of noise, and are implemented within other relevant works in the field [154, 155].

Amplitude damping models energy relaxation within a qubit that occurs through interactions with the environment over time. More information on the amplitude damping channel can be found in [30, 174]. Phase damping is a model of environmental noise that affects the representation of quantum information, without changes being made to the status of excitation or energy within the qubit itself. Phase damping can be modelled by the following Kraus operators, where $\lambda \in [0, 1]$ is the probability of qubit phase damping:

$$K_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix} \quad , \quad k_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix} \tag{6.5}$$

The application of Kraus operator K_0 does not affect the $|0\rangle$ portion of the quantum state, however negatively impacts the $|1\rangle$ portion by reducing its amplitude. This is the same operator that is used as a part of amplitude damping also, however the second Kraus operator K_1 is different. The application of K_1 affects the qubit by removing the $|0\rangle$ portion of the quantum state completely, as well as reducing the amplitude of the $|1\rangle$ portion alongside this. More information the phase damping channel can also be found in section 5.3.3 and [30].

For the following results, a subset of the MNIST dataset was used. This subset consisted of 250 training images per class used and 100 test images per class used. For each experiment, 30 epochs of optimization were conducted using the Adam optimizer with a learning rate of 10^{-4}. These hyperparameter values were selected as they were used throughout previous experiments conducted with MNIST data, and so consistency between experiments was desired.

The behaviour that is expected within these groups of experiments is that as the noise magnitude λ is increased, the general loss value would increase and the accuracy value would decrease in comparison to each experiment contained within the task nature (i.e., binary or multiclass classification task). However, as described previously within this chapter, it would be expected for a like-for-like value to produce a lower performance score as more classes are introduced to the task.

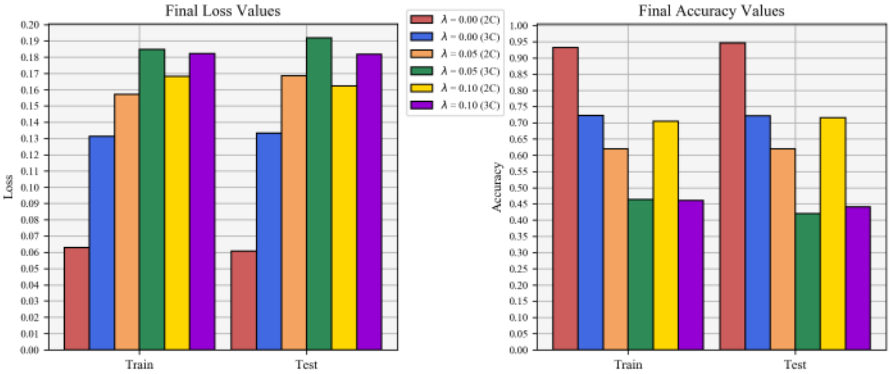


Figure 6.10: Final loss and classification accuracy values obtained through various binary (2C) and 3-class (3C) experiments conducted in a noisy simulation environment using the amplitude damping channel. The experiments were conducted using a subset of the MNIST dataset (classes of 0, 1 and 2), a single filter of size 3x3, and λ values of 0.05 and 0.1. For clarity, the experiment with zero noise is a direct reference to the experimental result displayed in table 6.1 and figure 6.2.

Charts displayed in figure 6.10 show experimental results obtained with the implementation of amplitude damping channels after each unitary operation, using qubit decay probabilities of 0.05 and 0.1 compared against previously gathered results with zero noise influence. The values for train set and test set loss and accuracy were taken after 30 epochs of optimization had been conducted.

The results displayed in figure 6.10 appear to follow the behaviour that is expected to some extent. For binary class experiments, a sharp decrease in loss is seen when noise is introduced, before a very slight loss once the noise magnitude is doubled. This does not translate across to accuracy values however, where the classification accuracy with $\lambda = 0.1$ is higher than that with $\lambda = 0.05$. This suggests that decay within the excitation status of the qubit affects the classification performance somewhat. However, once the impact of this is present, further reductions in performance are not in proportion to the magnitude of qubit decay.

The experiments using 3 classes of data also support this, as there is a significant increase in loss and decrease in accuracy as noise is introduced. However, these values appear to be very similar for $\lambda = 0.05$ and $\lambda = 0.1$, at an approximate loss of 0.184 and approximate accuracy of 45%. Overall, the system is affected to an extent by the introduction of qubit decay via an amplitude damping channel. Whilst initially this drop in performance is quite significant, the impact of noise with a greater magnitude is reduced.

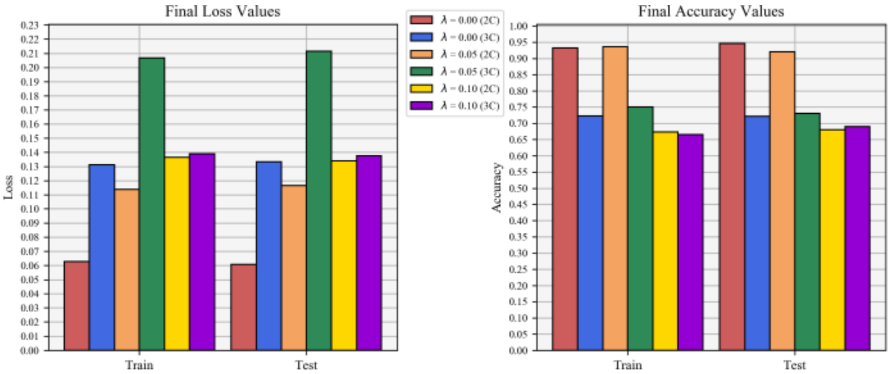


Figure 6.11: Final loss and classification accuracy values obtained through binary (2C) and 3-class (3C) experiments conducted in a noisy simulation environment using the phase damping channel. The experiments were conducted using a subset of MNIST data (classes of 0, 1 and 2), a single filter of size 3x3, and λ values of 0.05 and 0.1. For clarity, the experiment with zero noise is a direct reference to the experimental result displayed in table 6.1 and figure 6.2.

Figure 6.11 displays experimental result charts obtained through implementation of a phase damping channel after each unitary operation, using qubit damping probabilities of 0.05 and 0.1 as a comparison against the previously obtained result with zero noise influence. The values for train set and test set loss and accuracy were taken after 30 epochs of optimization had been conducted.

From the charts displayed in figure 6.11, the expected is followed for the most part. As the noise value λ increases, the loss values also increase for each task in turn. Similar to the behaviour exhibited by the loss values, the final classification accuracy values also follow the behaviour that would be expected somewhat. Within this, the only experiment which does not follow this pattern is where $\lambda = 0.05$ when using 3 classes (the green bar). Here there is a spike to the loss value, however the accuracy obtained is still comparable to similar experimental setups.

An explanation for this behaviour is that the embedding that the method has performed on the dataset results in datapoints being scattered around the borders between the three class regions. Even if a datapoint lies just within that class zone, it will be classified as such yet may still possess a larger distance between the ideal target state and itself. Over the course of the entire dataset for that epoch, this can equate to a larger value of loss for many datapoints located close to these boundaries, therefore it is difficult to label this experiment as an outlier and instead could be thought of as a difference in embedding.

Starting from experiments conducted with zero noise, there is a large initial increase in loss as noise is introduced to the simulation. However, as the noise value is doubled to 0.10, the increase in loss does not increase proportionally. Interestingly, whilst there is a large increase in loss here, this does not translate across to the classification accuracy values where the performance is comparable overall. As before, this could be explained by a more optimal distribution of initial weightings for the experiments with $\lambda = 0.05$, embedding datapoints within their correct class region more often than with zero noise. Or, another implication suggested here is that the system may exhibit a small amount of robustness against a lower-level of phase damping impact within the qubit.

Regardless of whether the loss value increases, suggesting the classification confidence is lower overall, the accuracy is maintained until noise magnitude is increased. To support this, the effect that phase damping has was even less for classification accuracy of the 3-class task, where performance is comparable within approximately 10% and 5% for train set and test set respectively as λ was increased. This is a promising factor to consider, as an innate robustness towards any kind of environmental noise can aid in optimization. In a case that the system was in a state capable of achieving 90+% classification accuracy on a 3+ class task using a single qubit, then any robustness held will be supportive to optimization if applied in a noisy quantum environment.

6.4 Discussion

To summarize the findings discussed in section 6.3 in its entirety, initial experimental results have been displayed that are obtained using a variety of datasets and applications from image classification to facial recognition. Overall, promising results have been achieved, given the purposes of the research and the system setup conditions posed. However, there are also key areas which would perhaps greatly benefit from further development and enable the performance of the system to be enhanced further.

In the case of binary classification experiments conducted using the MNIST data, the accuracy values obtained are not necessarily as high as the leading deep learning methodologies. However, the fact that the proposed method was able to reach test set accuracy scores in the 90% region within 5 epochs is promising in itself. Whilst realizing that the experiments conducted here only contained a subset of the MNIST data, and not the full dataset, it can be expected that the classification performance of the method will naturally drop slightly as the number of classes are increased, as well as the size of the dataset.

This was noticed following experiments conducted using 3 classes of the MNIST data, where classification accuracy stagnated at a lower value, and was unable to reach the high accuracy levels that would be desired in an image classification algorithm, such as well within the 90th percentile. It is to be noted that only a single filter containing 6 parameters was implemented over the course of this chapter, therefore complexity of the system can be increased by adding any number of filters to the experimental framework.

Noting that the MNIST dataset can sometimes be considered basic, or not truly representative of the classification capability of an algorithm, experiments were conducted using a subset of the Fashion-MNIST dataset to increase the 'difficulty' of the classification task. Again, the system showed promising results, reaching its highest classification accuracy values very close to 90% for a filter size of 3x3. In the context of this research these results are considered good, and show potential for the method to be enhanced further. As the complexity of embedding is increased, these results could be improved upon, allowing for the plateau in loss to be reduced to a much lower value.

In order to further demonstrate the initial capability of the proposed method, experiments using a bespoke dataset consisting of AT&T facial image data combined with CIFAR10 images were conducted. As with previous experiments, the results were not state-of-the-art, but considered promising and good in the context of the research and the experimental framework used. When applied to an additional secondary task of facial recognition using AT&T facial image data only, the system was unable to meet a satisfactory convergence to the data provided. As described previously, this is likely due to the small-scale data provided giving a lack of representation across the dataset, meaning the methodology was unable to learn and optimize effectively.

As the system was introduced to different environmental noise channels, initial results modelled using an amplitude damping channel suggest that noise greatly influences the qubit and reduces classification performance. However, when modelled using a phase damping channel, initial results appear to suggest a lack of impact or a slight robustness against the effect that phase damping has by manipulating the datapoint embeddings. As noise levels were increased higher, a subsequent drop in classification accuracy could be seen.

Whilst this could be seen as a negative point considering the current NISQ era of quantum computation, it is common to see this drop in classification performances across many quantum algorithms when noise is introduced [144]. With further development, it is hoped that any potential robustness can realized, or improved upon to enhance performance when applied in noisy environments. A speculative suggestion here may be to investigate whether applying additional filters may mimic the effect of data re-uploading, which is suggested to improve expressivity within the qubit [180], and thus may provide some robustness to noisy environments with additional layers, in particular the amplitude damping channel [174]. Exploring modifications such as this may aid in the robustness of the proposed method, and perhaps decrease any drop in classification such as that seen in figure 9 within noisy environments.

To once again put the experiments conducted into perspective, the classification performance for each set of experimental results was able to be achieved using just 6 parameters in total. As the field of deep learning has progressed from relatively shallow [4] to very deep networks consisting of many thousands of parameters [6], it should be considered here that the research being presented is proposed as a foundation, or starting point to progress forwards from. As has occurred for many modern ML algorithms, modifications and adaptations need to occur to improve upon previous performance and meet the task at hand. To that extent, there are a few notable ways where this research could be extended to provide additional insight and analysis into the feasibility of the algorithm as a quantum image classifier.

Firstly, an aspect well noted throughout this chapter is that of the low number of filters, and subsequently parameters optimized in this implementation. Whilst the point of this research was to showcase the potential with this few a number, it also opens a channel for further developments to remain efficient. Here, an analysis could be performed using additional numbers of filters to determine any difference in classification performance, and resulting embedding capability. In addition, a usage of localized weights as described in section 6.2 may continue to provide an advantage of maintaining spatial relationships between pixel values, without needing to increase the overall amount of implemented filters.

Following this, a secondary route for extension could envisage considering the inclusion of colorized images, to match a traditional image classification task specification more closely than focusing on greyscale images primarily. Within this, avenues to assign colour channels to individual qubits, as well as analyzing the effect of various entanglement operations between qubits may allow for a better understanding of how the methodology may extend to modern day tasks that include large-scale, colour images.

Finally, it is well noted that a significant limitation of a single qubit is the capability to classify many classes of data. As more classes of data are added, the subsequent area within the Bloch sphere that corresponds to each class is reduced. The effects of this reduction are much greater when there are a lower number of classes to begin with, however the ability to embed many datapoints into a very small section of the Bloch sphere will be difficult. Therefore, naturally we will be required to investigate the use of multiple qubits in order to contain sufficiently sized class boundaries when many classes are used. However, the point at which a single qubit is unable to cope with the number of classes is unknown. This point will also be undoubtedly affected by other factors such as the dimensionality or complexity of the data, as well as by factors that affect embedding complexity, such as re-uploading of data encodings seen in [125, 174].

6.5 Conclusion

In this chapter, a framework for efficient quantum image classification was proposed, requiring a minimum value of 6 parameters using a single qubit only. Multiple experiments were conducted using datasets of changing nature and difficulty to explore a variety of experimental results, and depth to our analysis. Initially, the results discussed throughout are promising, and display potential for the methodology to perform highly using a low number of parameters. The system was consistently able to achieve classification accuracy values within the 80th and 90th percentile in a short optimization timeframe within 30 training epochs.

However, when our experimental setup was applied to a noisy quantum simulation using amplitude damping and phase damping channels, classification accuracy was greatly reduced by the impact of qubit decay through amplitude damping. However, experimental results suggested a limited amount of robustness for classification performance against the impact of the phase damping channel by changes to the phase value of the qubit.

With experimental results considered, the proposed methodology provides a solid foundation to progress forwards to develop and build upon the success seen here using the bare minimum parameter and qubit count. As outlined in section 6.4, considerations for future research include an investigation into implementing additional filters, to determine whether classification performance can be improved upon and robustness similar to other works can be achieved. Alongside this, there are various opportunities to extend the foundational methodology proposed here towards modern-day image classification tasks that utilize high-resolution, colour images. These opportunities could examine the use of localized pixel weighting rather than individual filter weights, as well as investigate the effect of applying multiple qubits and entanglement measures to the system framework.

Overall, by extending the research presented in chapter 5, a QML-based image classification algorithm is presented, that greatly reduces the required parameter count from previous research, whilst maintaining pixel relationships that can assist in classification. It is interesting to find that a relatively high standard of classification can be performed with such few parameters and qubits, even within noisy quantum environments. This is a particularly important factor to consider in the NISQ era, which can make the proposed method of classification more favourable to employ over alternative QML algorithms that require many cohesive qubits, or large variational circuit structures of many parameters.

Chapter 7

Conclusion

7.1 Summary Of Thesis

Image classification is a highly-sensitive and challenging domain that gathers a tremendous amount of interest and development within ML research. However, major challenges that have arisen within the past decade include a growing complexity of deep CNN algorithms, stemming from the reliance on high-powered GPU units that are capable of processing large quantities of data quickly. As time has passed, hardware improvements are slowing down, meaning that we are edging closer to realising the potential of central and graphics processing units (CPU and GPU respectively). Therefore an alternative solution to implementing deeper or wider CNNs resulting in more weight parameters needs to be sought after.

QML is a growing field that aims to utilise natural quantum-mechanical principles within ML in order to provide an advantage of some sort over current classical ML algorithms. Although QML can be divided into 4 sub-categories, the CQ (classical-quantum) category is arguably the closest derivation to classical ML in the sense of the data and algorithm framework used. Typically, CQ algorithms consist of a process to encode classical data into its quantum form, followed by execution of a defined quantum circuit, followed by measurement and post-processing.

This thesis investigated the transition from classical-based to quantum-based solutions to counter the growing complexity problem faced within deep learning-based image classification. Typically, for very deep classical CNNs the vast majority of parameters lie within the feature extraction portion that contains the convolutional and pooling layers. Therefore, the first and only classical-based solution that was examined was the technique of transfer learning. Experimental results showed that, for a deep object detection framework with a CNN foundation, a high-level of classification accuracy could be maintained via transfer learning, reducing the computational load that training completely from scratch brings. However, it was noted that deep CNNs still require large quantities of computational power to process, therefore development towards more resource-efficient algorithms should be pursued.

After a review of previous literature, it is highlighted that there are 2 primary performance advantages that can be gained from quantum-based algorithms. The first performance advantage comes from operations that are naturally faster to compute quantumly versus classically, for example inner product estimations via a swap test subroutine or an unstructured search via Grovers algorithm. The second advantage to be gained is the formation of a naturally complex feature Hilbert space that is difficult to simulate classically. However, it remains an open question as to whether there exists an optimal strategy of encoding which is both sufficiently complex, yet remains efficient to do so.

Subsequent research addressed the first point by investigating a fidelity-based approach of classifiation directed towards the nature of images specifically. Fidelity is an approach often highlighted in literature as it is naturally efficient to calculate, therefore providing a computational speedup advantage. Typically, fidelity is implemented alongside a minimum search subroutine for a quantum k-nearest neighbour type algorithm. Therefore, the question remains as to the classification capability that fidelity calculation can have when implemented independently of other operations.

In order to address the second point, a review of various encoding strategies was conducted where a proposed scheme of single-qubit encoding was highlighted. This encoding strategy, which also incorporated a fidelity-based approach for measurement, was analysed in detail. Extending upon the in-depth analysis of single-qubit encoding, a novel quantum image classification methodology was proposed with the intent of remaining efficient in nature whilst maintaining a high embedding complexity.

To summarise, the key contributions of this thesis include in order of impact:

- The proposal of a novel quantum image classification algorithm based on efficient single-qubit encoding. The proposed algorithm maintained spatial relationships between pixels, whilst reducing the parameter count to a lower bound of 6, therefore improving efficiency.

- An enhanced understanding of single-qubit encoding towards the creation of complex yet robust feature spaces was gained. It was determined that depth values of 1-3 encoding layers provided the highest classification performance for the computation required. It was also demonstrated that improved robustness to environmental noise could be achieved with subsequent layers added.

- The demonstration of computational speedup gained via quantum operations versus similar classical methods. It was demonstrated that reasonable classification performance could be achieved via fidelity estimation alone. In addition, a computational speedup was reported over classical SVM and k-NN algorithms.

7.2 Research Findings And Discussion

This section presents the findings of research conducted within this thesis, and proposes concluding remarks in relation to the research objectives and questions outlined in section 1.2:

Research Question 1 *To what extent can the application of transfer learning maintain classification performance of deep CNNs, whilst reducing computational overhead from the optimization of large parameter counts?*

Chapter 3 aims to answer research question 1, and is the only working chapter that is completely classical-based in its computational format. Within this chapter, the transfer learning was implemented using the DL architecture YOLOv2. This leaves only the final fully-connected portion to be optimized to previously unseen data, rather than the entire parameter base of the architecture. Naturally this reduces the optimization time required per epoch, however it is questionable whether the application of transfer learning can maintain a high level of classification performance.

To investigate this, a set of experiments were conducted specifically towards an indoor-based object detection task. The experimental results gathered determined that a high level of classification accuracy was achieved approximately half-way through the alloted optimization timeframe, prior to a slight regression in performance. This is a positive insight towards transfer learning by requiring a lower epoch number to optimize fully, rather than training completely from scratch which may take a larger number of epochs to achieve a similar result.

The experimental results also showed that a high extent of classification performance could be maintained via the use of transfer learning. Overall, the implementation of this technique meant that a large proportion of transformation layers (i.e. convolutional/pooling layers), and subsequently a large share of the parameter count could be left frozen, negating the need for computationally expensive backpropagation operations whilst keeping a fair level of classification performance.

Research Question 2 *To what extent can a non-gradient based approach be taken to perform effective quantum image classification?*

A methodology was presented in chapter 4 that utilised a fidelity-based approach to classification in order to address this research question. A fidelity-based approach was chosen with the intention of improving computational speed, as it is a naturally efficient calculation for quantum computers to process and avoided time-consuming optimization of parameter sets. Experimental results demonstrated that a reasonable level of classification accuracy could be achieved, with very little in terms of quantum circuit operations. Alongside this, the methodology achieved a computational speedup over classical ML algorithms of an SVM and k-NN. Additional experimental results contained within the study suggest that qubit management may be a key factor in determining effectiveness of the developed algorithm.

The research conducted here demonstrates that a non-gradient based classification approach using fidelity estimation is able to produce effective classification capability to a certain extent. An advantage of taking this approach for classification is the removal of the need for parameter optimization, which can be extremely time consuming for even moderately sized systems. Alongside this, the additional implementation of an oracle routine, which was often seen throughout many works that utilise the metric of fidelity, was disregarded. Although this meant that a single circuit execution was required per image compared, it negated the storage of an entire dataset in a superposition state, requiring many qubits. Arguably, for very small datasets containing few small-sized images, few qubits may be required to store the complete set of data. However, for the context of NISQ quantum computation it was felt that this trade-off was appropriate to reduce the possibility of errors that can occur as a result of poor qubit cohesion.

Research Question 3 *Can effective feature representations of classical data be created in Hilbert space using low numbers of quantum bits?*

In order to address this research question, the modern development of single-qubit encoding was analysed in a classification setting. Experimental results determined that the sharpest of classification improvements were gained when depth values of 1-3 uploading layers were implemented. However, the implementation of additional layers did not provide a substantive classification performance improvement. Alongside this, experiments were conducted within a noisy simulation environment through the addition of the amplitude damping noise channel. Here, experimental results suggested that a level of robustness could be gained by the implementation of increased depth from additional uploading layers. As before, the sharpest robustness advantages could be gained when 1-3 uploading layers were used.

The research conducted here determines that an effective feature representation can be created with a single qubit implemented only. For much of the literature surrounding quantum machine learning algorithms, typically a basis or amplitude encoding approach is taken to embed classical data into Hilbert space. Although logarithmically efficient in nature, the capability to encode data through parametrized rotations onto a single qubit is particularly useful for minimizing the amount of quantum resources required for execution.

Research Question 4 *Can a novel quantum machine learning classifier be developed for image classification that makes efficient use of resources provided?*

In order to address research question 4, a novel QML classifier was developed in a manner tailored towards image classification tasks. Inspired by convolutional operations, the encoding strategy was adapted in order to maintain spatial relationships between pixels, therefore assigning each filter with a small set of 6 parameters only. For analysis of the effectiveness of the proposed algorithm, a group of experiments were conducted using various image datasets. For all experiments, a single filter was implemented that required a set of 6 parameters in total. The experimental results contained within chapter 6 showed promising classification capabilities for the proposed algorithm. Alongside this, the proposed algorithm demonstrated an extent of robustness against the implementation of the phase damping environmental noise channel.

The experimental results obtained determine that effective quantum-based image classification is capable of being performed, whilst making effective use of low resources available with a single qubit. Although the experimental results obtained are not state-of-the-art for the data used, considering the low parameter count and that each pixel was to be encoded once only the initial classification capability is extremely promising.

7.3 Reflection And Recommendation For Future Work

Overall, the research contained within this thesis has many implications that have been discussed previously. Naturally, this opens up various routes for future research, therefore to finalise this section it is relevant to outline and discuss these considerations towards avenues for future development.

- A limitation that should be noted for the method implemented in chapter 4 is how the inclusion of multi-class data is conducted. A suggestion made is to perhaps include a circuit execution per data class, therefore classifying in a one-versus-all manner. However, it remains an open question as to how this can be extended to tasks of a larger scale and difficulty. Therefore, future development of this method may be directed towards how difficult image data may be classified in this manner. For image classes that are very inconsistent in object features such as location within the image, aspect ratio, or contrast, then fidelity as a classification tool may prove more challenging than for images like the format used here. Therefore, further investigation towards these areas may enhance the overall capability and robustness of the algorithm itself.

- The quantum image classification algorithm proposed in chapter 6 presents a vast array of avenues for further investigation and development. Firstly, the experimental setup implemented a single filter only of 6 weight parameters. This was done intentionally to show the baseline capability with as minimal parameterization and operations as possible. However, investigation towards benefits that the implementation of additional filters could bring may enhance the working knowledge of the proposed algorithm.

- Finally, it remains an open question as to how the colour values of each pixel may be encoded. A suggestion to how this may be performed could be via a multi-qubit implementation, such as assigning values for a particular colour to an individual qubit then encoded as default. Another supporting factor for a multi-qubit approach is the extension to tasks that involve a higher number of classes. Additional qubits would allow for a greater dimension of Hilbert space, therefore providing a greater area for each individual class boundaries. However, the way in which the extension to multiple qubits is implemented in practice is largely undecided. Logically, there would need to be some form of entanglement to provide an element of dependency between the implemented qubits. This aspect opens up a secondary area for investigation, to analyse whether a particular entanglement setup may provide advantages over others.

7.4 Concluding Statement

This thesis implemented, analysed and developed a set of classical-based and quantum-based machine learning techniques and algorithms targeted towards the domain of image classification. The underlying intentions of this thesis are to improving the efficiency of image classification algorithms, whilst maintaining a high degree of classification accuracy. The following concluding remarks summarise the key research points as a whole:

- Transfer learning remains a viable technique to consider in order to reduce computational load during optimization of deep CNNs for image classification and object detection. Experimental results demonstrated that a good classification accuracy result of 91.47% was achieved, whilst removing the optimization requirement for 76.9% of the total parameterized layers.

- The removal of costly backpropagation was considered for quantum-based image classification via the implementation of a fidelity-based routine. Classification was based on similarity between a dataset instance and a defined comparison vector. Experimental results determined that fidelity alone produced a classification accuracy of 90.6% for the defined dataset. Notably, the implemented method improved computational complexity over similar classical algorithms through utilising resource-efficient amplitude encoding.

- A novel quantum image classifier was proposed based on the single-qubit encoding scheme. Initial analysis of single-qubit encoding showed it possessed the capability to encode classical data into a suitably-complex feature space. The greatest performance improvements were also seen to occur within 1-3 data encoding layers. The proposed system modified the structure of single-qubit encoding, in order to maintain pixel relationships and greatly reduce the parameter count, with itent of shortening computational load during optimization. Experimental results determined that the proposed method had great potential as a quantum image classification tool, whilst remaining efficient in its use of quantum resources and robust to certain environmental noise impacts.

References

[1] Yuanqing Lin et al. "Large-scale image classification: Fast feature extraction and SVM training". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2011), pp. 1689–1696. ISSN: 10636919. DOI: 10.1109/CVPR.2011.5995477.

[2] Neha Sharma, Vibhor Jain, and Anju Mishra. "An Analysis of Convolutional Neural Networks for Image Classification". In: *Procedia Computer Science* 132 (2018), pp. 377–384. ISSN: 18770509. DOI: 10.1016/j.procs.2018.05.198. URL: www.sciencedirect.com.

[3] Kaiming He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 International Conference on Computer Vision, ICCV 2015 (2015), pp. 1026–1034. ISSN: 15505499. DOI: 10.1109/ICCV.2015.123. arXiv: 1502.01852.

[4] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2323. ISSN: 00189219. DOI: 10.1109/5.726791.

[5] "Learning algorithms for classification: A comparison on handwritten digit recognition". In: *Neural networks: the statistical mechanics perspective* (1995), pp. 261–276.

[6] Yahia Assiri. "Stochastic Optimization of Plain Convolutional Neural Networks with Simple methods". In: (2020). arXiv: 2001.08856. URL: http://arxiv.org/abs/2001.08856.

[7] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 15731405. DOI: 10.1007/s11263-015-0816-y. arXiv: 1409.0575. URL: http://image-net.org/challenges/LSVRC/.

[8] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (2021), pp. 43–76. ISSN: 15582256. DOI: 10.1109/JPROC.2020.3004555. arXiv: 1911.02685.

[9] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum principal component analysis”. In: *Nature Physics* 10.9 (July 2014), pp. 631–633. ISSN: 17452481. DOI: 10.1038/NPHYS3029. arXiv: 1307.0401.

[10] C. M. Wilson et al. *Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers.* Tech. rep. 2018. arXiv: 1806.08321. URL: http://arxiv.org/abs/1806.08321.

[11] Vojtěch Havlíček et al. *Supervised learning with quantum-enhanced feature spaces.* Tech. rep. 7747. 2019, pp. 209–212. DOI: 10.1038/s41586-019-0980-2. arXiv: 1804.11326.

[12] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. *Quantum convolutional neural networks.* Tech. rep. 12. 2019, pp. 1273–1278. DOI: 10.1038/s41567-019-0648-8. arXiv: 1810.03787.

[13] “Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks”. In: *Journal of Physics: Conference Series* 1471.1 (2020), p. 12010. ISSN: 17426596. DOI: 10.1088/1742-6596/1471/1/012010.

[14] Wei Jiang et al. “Multilayer Perceptron Neural Network for Surface Water Extraction in Landsat 8 OLI Satellite Images”. In: *Remote Sensing 2018, Vol. 10, Page 755* 10.5 (May 2018), p. 755. DOI: 10.3390/RS10050755. URL: https://www.mdpi.com/2072-4292/10/5/755/htm%20https://www.mdpi.com/2072-4292/10/5/755.

[15] Radmila Janković. “Machine learning models for cultural heritage image classification: Comparison based on attribute selection”. In: *Information (Switzerland)* 11.1 (2020). ISSN: 20782489. DOI: 10.3390/info11010012. URL: https://old.datahub.io/dataset/architectural-heritage-elements-image-.

[16] Sarfraz Nawaz Brohi et al. “A Multilayer Perceptron Model for the Classification of Breast Cancer Cells”. In: *International Journal of Computing and Digital Systems* (2021). URL: http://journals.uob.edu.bh.

[17] S. Ben Driss et al. “A comparison study between MLP and convolutional neural network models for character recognition”. In: *Real-Time Image and Video Processing 2017* 10223 (2017), p. 1022306. ISSN: 1996756X. DOI: 10.1117/12.2262589. URL: https://hal-upec-upem.archives-ouvertes.fr/hal-01525504.

[18] Christian Szegedy et al. “Going Deeper with Convolutions”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June-2015 (Sept. 2014), pp. 1–9. arXiv: 1409.4842. URL: https://arxiv.org/abs/1409.4842v1.

[19] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Dec. 2015), pp. 770–778. arXiv: 1512.03385. URL: https://arxiv.org/abs/1512.03385v1.

[20] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (Aug. 2016), pp. 2261–2269. arXiv: 1608.06993. URL: https://arxiv.org/abs/1608.06993v5.

[21] Christian Szegedy et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017* (Feb. 2016), pp. 4278–4284. arXiv: 1602.07261. URL: https://arxiv.org/abs/1602.07261v2.

[22] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Nov. 2013), pp. 580–587. arXiv: 1311.2524. URL: https://arxiv.org/abs/1311.2524v5.

[23] Ross Girshick. “Fast R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2015 International Conference on Computer Vision, ICCV 2015 (2015), pp. 1440–1448. ISSN: 15505499. DOI: 10.1109/ICCV.2015.169. arXiv: 1504.08083. URL: https://github.com/rbgirshick/.

[24] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2015), pp. 1137–1149. arXiv: 1506.01497. URL: https://arxiv.org/abs/1506.01497v3.

[25] Kaiming He et al. “Mask R-CNN”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Mar. 2017), pp. 386–397. arXiv: 1703.06870. URL: https://arxiv.org/abs/1703.06870v3.

[26] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. “DensePose: Dense Human Pose Estimation In The Wild”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Feb. 2018), pp. 7297–7306. arXiv: 1802.00434. URL: https://arxiv.org/abs/1802.00434v1.

[27] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (June 2015), pp. 779–788. arXiv: `1506.02640`. URL: `https://arxiv.org/abs/1506.02640v5`.

[28] EveringhamMark et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338. DOI: `10.1007/S11263-009-0275-4`. URL: `https://dl.acm.org/doi/abs/10.1007/s11263-009-0275-4`.

[29] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (Dec. 2016), pp. 6517–6525. arXiv: `1612.08242`. URL: `https://arxiv.org/abs/1612.08242v1`.

[30] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information.* Vol. 27. 3. ISBN: 9781107002173. URL: `www.cambridge.org`.

[31] Diogo A.C. Narciso and F. G. Martins. "Application of machine learning tools for energy efficiency in industry: A review". In: *Energy Reports* 6 (Nov. 2020), pp. 1181–1199. ISSN: 23524847. DOI: `10.1016/j.egyr.2020.04.035`.

[32] Hao nan Guo et al. "Application of machine learning methods for the prediction of organic solid waste treatment and recycling processes: A review". In: *Bioresource Technology* 319 (Jan. 2021), p. 124114. ISSN: 18732976. DOI: `10.1016/j.biortech.2020.124114`.

[33] Muhammad Attique Khan et al. "Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists". In: *Diagnostics* 10.8 (Aug. 2020), p. 565. ISSN: 20754418. DOI: `10.3390/diagnostics10080565`. URL: `https://www.mdpi.com/2075-4418/10/8/565/htm%20https://www.mdpi.com/2075-4418/10/8/565`.

[34] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. "Machine learning in a quantum world". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4013 LNAI (2006), pp. 431–442. ISSN: 16113349. DOI: `10.1007/11766247_37`. URL: `https://link.springer.com/chapter/10.1007/11766247%7B%5C_%7D37`.

[35] Marvin Becker and Karl-Rehan Chiu Falck. "Deep Reinforcement Learning for Quantum Error Correction". In: *Chalmers tekniska högskola* (2020). URL: `https://odr.chalmers.se/handle/20.500.12380/302593`.

[36] Andrea Mari et al. “Transfer learning in hybrid classical-quantum neural networks”. In: *Quantum* 4 (Dec. 2020). ISSN: 2521327X. DOI: 10.22331/Q-2020-10-09-340. arXiv: 1912.08278. URL: http://arxiv.org/abs/1912.08278.

[37] Maria Schuld and Francesco Petruccione. *Quantum Science and Technology Supervised Learning with Quantum Computers*. Tech. rep. 2018. URL: http://www.springer.com/series/10039.

[38] Xu Ji, Andrea Vedaldi, and Joao Henriques. “Invariant information clustering for unsupervised image classification and segmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-October (2019), pp. 9864–9873. ISSN: 15505499. DOI: 10.1109/ICCV.2019.00996. arXiv: 1807.06653.

[39] Sungwon Park et al. “Improving Unsupervised Image Clustering With Robust Learning”. In: (2020). arXiv: 2012.11150. URL: http://arxiv.org/abs/2012.11150.

[40] Wouter Van Gansbeke et al. “SCAN: Learning to Classify Images Without Labels”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12355 LNCS (2020), pp. 268–285. ISSN: 16113349. DOI: 10.1007/978-3-030-58607-2_16. arXiv: 2005.12320. URL: www.github.com/wvangansbeke/.

[41] Gavin E. Crooks. “Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition”. In: (2019). arXiv: 1905.13311. URL: http://arxiv.org/abs/1905.13311.

[42] Gary G. Koch et al. “A General Methodology for the Analysis of Experiments with Repeated Measurement of Categorical Data”. In: *Biometrics* 33.1 (Mar. 1977), p. 133. ISSN: 0006341X. DOI: 10.2307/2529309.

[43] Donald W. Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. 1963. DOI: 10.1137/0111030. URL: https://www.jstor.org/stable/2098941 (visited on 10/09/2021).

[44] J. A.K. Suykens and J. Vandewalle. “Least squares support vector machine classifiers”. In: *Neural Processing Letters* 9.3 (1999), pp. 293–300. ISSN: 13704621. DOI: 10.1023/A:1018628609742. URL: https://link.springer.com/article/10.1023/A:1018628609742.

[45] Chin Wynne W. *The partial least squares approach for structural equation modeling. - PsycNET*. 1998. URL: https://psycnet.apa.org/record/1998-07269-010 (visited on 10/09/2021).

[46] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for solving linear systems of equations". In: *Physical Review Letters* 103.15 (Nov. 2008). DOI: 10.1103/physrevlett.103.150502. arXiv: 0811.3171. URL: https://arxiv.org/abs/0811.3171v3.

[47] Nathan Wiebe, Daniel Braun, and Seth Lloyd. "Quantum algorithm for data fitting". In: *Physical Review Letters* 109.5 (Aug. 2012), p. 050505. ISSN: 00319007. DOI: 10.1103/PhysRevLett.109.050505. URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.109.050505.

[48] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. *Simulating a perceptron on a quantum computer.* Tech. rep. 7. 2015, pp. 660–663. DOI: 10.1016/j.physleta.2014.11.061. arXiv: 1412.3635.

[49] Guoming Wang. "Quantum algorithm for linear regression". In: *Physical Review A* 96.1 (2017). ISSN: 24699934. DOI: 10.1103/PhysRevA.96.012335. arXiv: 1402.0660.

[50] Yang Liu and Shengyu Zhang. "Fast quantum algorithms for least squares regression and statistic leverage scores". In: *Theoretical Computer Science* 657 (Jan. 2017), pp. 38–47. ISSN: 03043975. DOI: 10.1016/j.tcs.2016.05.044.

[51] Chao-Hua Yu, Fei Gao, and Qiaoyan Wen. *An improved quantum algorithm for ridge regression.* Tech. rep. 1. 2019, pp. 1–1. DOI: 10.1109/tkde.2019.2937491. arXiv: 1707.09524.

[52] Arthur E Hoerl and Robert W Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". In: *Source: Technometrics* 12.1 (1970), pp. 55–67.

[53] D Anguita et al. "K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines". In: *International Conference on Data Mining* June 2014 (2009), pp. 291–297.

[54] André R. de Geus, André R. Backes, and Jefferson R. Souza. "Variability evaluation of CNNs using cross-validation on viruses images". In: *VISIGRAPP 2020 - Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* 4 (2020), pp. 626–632. DOI: 10.5220/0009352106260632. URL: www.cb.uu.se/.

[55] "Evaluation of the K-Nearest Neighbor Model With K-Fold Cross Validation on Image Classification". In: *Jurnal Infokum* 9.1 (Dec. 2020), pp. 1–6. ISSN: 2722-4635. URL: http://seaninstitute.org/infor/index.php/infokum/article/view/72.

[56] András Gilyén, Zhao Song, and Ewin Tang. "An improved quantum-inspired algorithm for linear regression". In: (2020). arXiv: 2009.07268. URL: http://arxiv.org/abs/2009.07268.

[57] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. *Quantum support vector machine for big data classification.* Tech. rep. 3. 2014. DOI: 10.1103/PhysRevLett.113.130503. arXiv: 1307.0471.

[58] Padraig Cunningham and Sarah Jane Delany. "k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)". In: (2020). DOI: 10.1145/3459665. arXiv: 2004.04523. URL: http://arxiv.org/abs/2004.04523%7B%5C%%7D0Ahttp://dx.doi.org/10.1145/3459665.

[59] Arif Ridho Lubis, Muharman Lubis, and Al-Khowarizmi. "Optimization of distance formula in k-nearest neighbor method". In: *Bulletin of Electrical Engineering and Informatics* 9.1 (Feb. 2020), pp. 326–338. ISSN: 23029285. DOI: 10.11591/eei.v9i1.1464.

[60] Aman Kataria and M D Singh. "A Review of Data Classification Using K-Nearest Neighbour Algorithm". In: *International Journal of Emerging Technology and Advanced Engineering* 3.6 (2013), pp. 354–360. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.413.3893.

[61] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. *Quantum algorithms for supervised and unsupervised machine learning.* Tech. rep. 2013. arXiv: 1307.0411. URL: http://arxiv.org/abs/1307.0411.

[62] Brad Lackey. *Introduction to quantum information processing.* 2016. URL: https://www.cs.umd.edu/class/fall2016/cmsc858K/LectureNotes/Fidelity.pdf (visited on 10/10/2021).

[63] D J Kok. "Building a quantum kNN with Qiskit : theoretical gains put to practice". In: (2021). URL: https://www.ru.nl/publish/pages/913395/daniel%7B%5C_%7Dkok%7B%5C_%7D4%7B%5C_%7Dmaart%7B%5C_%7D2021.pdf.

[64] Yijie Dang et al. *Image classification based on quantum K-Nearest-Neighbor algorithm.* Tech. rep. 9. 2018. DOI: 10.1007/s11128-018-2004-9. arXiv: 1805.06260v1.

[65] Christoph Durr and Peter Hoyer. "A quantum algorithm for finding the minimum". In: *arXiv preprint quant-ph/9607014* (1996).

[66] D Adu-Gyamfi et al. "Improved Handwritten Digit Recognition using Quantum K-Nearest Neighbor Algorithm". In: *Article in International Journal of Theoretical Physics* (2019). DOI: 10.1007/s10773-019-04124-5. URL: https://doi.org/10.1007/s10773-019-04124-5.

[67] Yue Ruan et al. "Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance". In: *International Journal of Theoretical Physics* 56.11 (2017), pp. 3496–3507. ISSN: 15729575. DOI: 10.1007/s10773-017-3514-4. URL: https://www.researchgate.net/publication/319251548.

[68] Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore. "Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning". In: *Quantum Information and Computation* 15.3-4 (2015), pp. 318–358. ISSN: 15337146. DOI: 10.26421/qic15.3-4-7. arXiv: 1401.2142.

[69] Edward Farhi and Hartmut Neven. *Classification with Quantum Neural Networks on Near Term Processors*. Tech. rep. 2018. arXiv: 1802.06002. URL: http://arxiv.org/abs/1802.06002.

[70] K. Mitarai et al. *Quantum circuit learning*. Tech. rep. 3. 2018. DOI: 10.1103/PhysRevA.98.032309. arXiv: 1803.00745.

[71] Maria Schuld and Nathan Killoran. *Quantum Machine Learning in Feature Hilbert Spaces*. Tech. rep. 4. 2019. DOI: 10.1103/PhysRevLett.122.040504. arXiv: 1803.07128.

[72] Maria Schuld et al. "Circuit-centric quantum classifiers". In: *Physical Review A* 101.3 (Apr. 2020). ISSN: 24699934. DOI: 10.1103/PhysRevA.101.032308. arXiv: 1804.00633. URL: http://arxiv.org/abs/1804.00633%20http://dx.doi.org/10.1103/PhysRevA.101.032308.

[73] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[74] Kerstin Beer et al. "Training deep quantum neural networks". In: *Nature Communications* 11.1 (2020). ISSN: 20411723. DOI: 10.1038/s41467-020-14454-2. URL: https://doi.org/10.1038/s41467-020-14454-2.

[75] Xiangyong Cao et al. "Hyperspectral Image Classification with Convolutional Neural Network and Active Learning". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.7 (2020), pp. 4604–4616. ISSN: 15580644. DOI: 10.1109/TGRS.2020.2964627. URL: https://www.researchgate.net/publication/339018804.

[76] Andrii O. Tarasenko, Yuriy V. Yakimov, and Vladimir N. Soloviev. "Convolutional neural networks for image classification". In: *CEUR Workshop Proceedings* 2546 (2019), pp. 101–114. ISSN: 16130073.

[77] S. H.Shabbeer Basha et al. "Impact of fully connected layers on performance of convolutional neural networks for image classification". In: *Neurocomputing* 378 (2020), pp. 112–119. ISSN: 18728286. DOI: 10.1016/j.neucom.2019.10.008. arXiv: 1902.02771. URL: https://doi.org/10.1016/j.neucom.2019.10.008.

[78] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* http://www.deeplearningbook.org. MIT Press, 2016.

[79] Gousia Habib and Shaima Qureshi. "Optimization and acceleration of convolutional neural networks: A survey". In: *Journal of King Saud University - Computer and Information Sciences* (Oct. 2020). ISSN: 22131248. DOI: 10.1016/j.jksuci.2020.10.004.

[80] Chih Ting Liu et al. "Computation-Performance Optimization of Convolutional Neural Networks with Redundant Kernel Removal". In: *Proceedings - IEEE International Symposium on Circuits and Systems* 2018-May (2018). ISSN: 02714310. DOI: 10.1109/ISCAS.2018.8351053. arXiv: 1705.10748.

[81] Wenquan Du, Zixin Wang, and Dihu Chen. "Optimizing of Convolutional Neural Network Accelerator". In: *Green Electronics* (Mar. 2018). DOI: 10.5772/intechopen.75796. URL: https://www.intechopen.com/chapters/60223.

[82] Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. *Quantum Algorithms for Deep Convolutional Neural Networks.* Tech. rep. 2019. arXiv: 1911.01117. URL: http://arxiv.org/abs/1911.01117.

[83] Pablo Antonio Moreno Casares. "Circuit implementation of bucket brigade qRAM for quantum state preparation". In: *arXiv* (2020). arXiv: 2006.11761. URL: http://arxiv.org/abs/2006.11761.

[84] Jeongho Bang et al. "Optimal usage of quantum random access memory in quantum machine learning". In: *Physical Review A* 99.1 (2019). ISSN: 24699934. DOI: 10.1103/PhysRevA.99.012326. arXiv: 1809.04814.

[85] Ryo Asaka, Kazumitsu Sakai, and Ryoko Yahagi. "Quantum random access memory via quantum walk". In: *Quantum Science and Technology* 6.3 (2021). ISSN: 20589565. DOI: 10.1088/2058-9565/abf484. arXiv: 2008.13365.

[86] Aaron Green and Emily Kaplitz. *Quantum Random Access Memory.* 2019. (Visited on 10/10/2021).

[87] Seunghyeok Oh et al. "Quantum Convolutional Neural Network for Resource-Efficient Image Classification: A Quantum Random Access Memory (QRAM) Approach". In: *International Conference on Information Networking* 2021-Janua (Jan. 2021), pp. 50–52. DOI: 10.1109/ICOIN50884.2021.9333906.

[88] Maxwell Henderson et al. *Quanvolutional neural networks: powering image recognition with quantum circuits*. Tech. rep. 1. 2020, pp. 1–9. DOI: 10.1007/s42484-020-00012-y. arXiv: 1904.04767.

[89] Alex Krizhevsky and G Hinton. "Learning multiple layers of features from tiny images.(2009)". In: *Cs.Toronto.Edu* (2009), pp. 1–58. URL: http://www.cs.toronto.edu/%7B~%7Dkriz/cifar.html.

[90] *World Population Ageing, United Nations, 2017*. 2017. URL: https://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017%7B%5C_%7DHighlights.pdf.

[91] D. W. Elmendorf. *The 2014 Long-Term Budget Outlook*. Tech. rep. DTIC, 2014.

[92] UNISON. *UNISON's Staffing Levels Survey*. 2015. URL: https://www.unison.org.uk/content/uploads/2015/04/%20TowebRedAlert-Unsafe-Staffing-Levels-Rising1.pdf.

[93] WHO. *WHO Global Report On Falls Prevention In Older Age*. Tech. rep. WHO, 2007. URL: https://extranet.who.int/agefriendlyworld/wp-content/uploads/2014/06/WHo-Global-report-on-falls-prevention-in-older-age.pdf.

[94] Jane Fleming and Carol Brayne. "Inability to get up after falling, subsequent time on floor, and summoning help: prospective cohort study in people over 90". In: *BMJ* 337.7681 (Nov. 2008), pp. 1279–1282. ISSN: 0959-8138. DOI: 10.1136/BMJ.A2227. URL: https://www.bmj.com/content/337/bmj.a2227%20https://www.bmj.com/content/337/bmj.a2227.abstract.

[95] Cummings SR, Rubin SM, and Black D. "The future of hip fractures in the United States. Numbers, costs, and potential effects of postmenopausal estrogen". In: *Clinical orthopaedics and related research* 252 (1990), pp. 163–166. ISSN: 0009-921X. URL: https://pubmed.ncbi.nlm.nih.gov/2302881/.

[96] Iffat Zafar et al. "Human silhouette extraction on FPGAs for infrared night vision military surveillance". In: *2010 2nd Pacific-Asia Conference on Circuits, Communications and System, PACCS 2010* 1 (2010), pp. 63–66. DOI: 10.1109/PACCS.2010.5627025.

[97] Chia-Yen Chiang et al. "Deep Learning based Automated Forest Health Diagnosis from Aerial Images". In: *IEEE Access* 8 (Oct. 2020), pp. 144064–144076. DOI: 10.1109/access.2020.3012417.. arXiv: 2010.08437. URL: https://arxiv.org/abs/2010.08437v1.

[98] Gary Storey et al. "3DPalsyNet: A facial palsy grading and motion recognition framework using fully 3D convolutional neural networks". In: *IEEE Access* 7 (May 2019), pp. 121655–121664. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2937285. arXiv: 1905.13607. URL: https://arxiv.org/abs/1905.13607v1.

[99] Richard Jiang and Danny Crookes. "Shallow Unorganized Neural Networks Using Smart Neuron Model for Visual Perception". In: *IEEE Access* 7 (2019), pp. 152701–152714. DOI: 10.1109/ACCESS.2019.2946422.

[100] Gary Storey, Richard Jiang, and Ahmed Bouridane. "Role for 2D image generated 3D face models in the rehabilitation of facial palsy". In: *Healthcare Technology Letters* 4.4 (2017), p. 145. DOI: 10.1049/HTL.2017.0023. URL: /pmc/articles/PMC5569939/%20/pmc/articles/PMC5569939/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5569939/.

[101] Ziping Jiang et al. "Social Behavioral Phenotyping of Drosophila with a 2D-3D Hybrid CNN Framework". In: *IEEE Access* 7 (2019), pp. 67972–67982. DOI: 10.1109/ACCESS.2019.2917000.

[102] Gary Storey, Ahmed Bouridane, and Richard Jiang. "Integrated Deep Model for Face Detection and Landmark Localization From "In The Wild" Images". In: *IEEE Access* 6 (2018), pp. 74442–74452. DOI: 10.1109/ACCESS.2018.2882227.

[103] Alisha Rege et al. "Vision-based approach to senior healthcare: Depth-based activity recognition with convolutional neural networks". In: *Semantic Scholar* (2017), pp. 1–8.

[104] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. "Vision-based fall detection with convolutional neural networks". In: *Wireless Communications and Mobile Computing* 2017 (2017). DOI: 10.1155/2017/9474806.

[105] Xiaogang Li et al. "Fall detection for elderly person care using convolutional neural networks". In: *Proceedings - 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2017* 2018-January (2018), pp. 1–6. DOI: 10.1109/CISP-BMEI.2017.8302004. URL: https://ieeexplore.ieee.org/abstract/document/8302004/.

[106] Zelun Luo et al. "Computer Vision-based Approach to Maintain Independent Living for Seniors". In: *AMIA*. 2017.

[107] Richard Jiang and Danny Crookes. "Deep salience: Visual salience modeling via deep belief propagation". In: *Proceedings of the National Conference on Artificial Intelligence* 4 (2014), pp. 2773–2779. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewPaper/8506.

[108] R Jiang, D Crookes - Twenty-Eighth AAAI Conference on Artificial, and undefined 2014. "Deep salience: Visual salience modeling via deep belief propagation". In: *aaai.org* (). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewPaper/8506.

[109] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *32nd International Conference on Machine Learning, ICML 2015* 1 (Feb. 2015), pp. 448–456. arXiv: 1502.03167. URL: https://arxiv.org/abs/1502.03167v3.

[110] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[111] Glenn Porter and Greg Doran. "An anatomical and photographic technique for forensic facial identification". In: *Forensic Science International* 114.2 (2000), pp. 97–105. ISSN: 03790738. DOI: 10.1016/S0379-0738(00)00290-5.

[112] Ghulam Muhammad et al. "A Facial-Expression Monitoring System for Improved Healthcare in Smart Cities". In: *IEEE Access* 5 (June 2017), pp. 10871–10881. ISSN: 21693536. DOI: 10.1109/ACCESS.2017.2712788.

[113] Ching Yiu Jessica Liu. "Facial Identification from Online Images for Use in the Prevention of Child Trafficking and Exploitation". In: *PQDT - Global* (2018), p. 186. URL: https://search.proquest.com/dissertations-theses/facial-identification-online-images-use/docview/2495344687/se-2?accountid=13042%7B%5C%%7D0Ahttp://oxfordsfx.hosted.exlibrisgroup.com/oxford?url%7B%5C_%7Dver=Z39.88-2004%7B%5C&%7Drft%7B%5C_%7Dval%7B%5C_%7Dfmt=info:ofi/fmt:kev:mtx:dissertation%7B%5C&%7Dgenre=disse.

[114] Radu Lixăndroiu et al. "Personalized Online Marketing Using Facial Recognition". In: *Journal of Smart Economic Growth* 5.2 (Dec. 2020), pp. 1–10. ISSN: 2537-141X. URL: https://jseg.ro/index.php/jseg/article/view/109.

[115] Samuel Yen-Chi Chen et al. *Variational Quantum Circuits for Deep Reinforcement Learning.* Tech. rep. 2019. arXiv: 1907.00397. URL: http://arxiv.org/abs/1907.00397.

[116] C. M. Wilson et al. *Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers.* Tech. rep. 2018. arXiv: 1806.08321. URL: http://arxiv.org/abs/1806.08321.

[117] Seth Lloyd et al. *Quantum embeddings for machine learning.* Tech. rep. 2020. arXiv: 2001.03622. URL: http://arxiv.org/abs/2001.03622.

[118] Nathan Killoran et al. "Continuous-variable quantum neural networks". In: *Physical Review Research* 1.3 (2019), p. 33063. DOI: 10.1103/physrevresearch.1.033063. arXiv: 1806.06871.

[119] Gregory R. Steinbrecher et al. "Quantum optical neural networks". In: *npj Quantum Information* 5.1 (2019). ISSN: 20566387. DOI: 10.1038/s41534-019-0174-7. arXiv: 1808.10047. URL: https://doi.org/10.1038/s41534-019-0174-7.

[120] Siddhant Garg and Goutham Ramakrishnan. *Advances in Quantum Deep Learning: An Overview.* Tech. rep. 2020. arXiv: 2005.04316. URL: http://arxiv.org/abs/2005.04316.

[121] Afham, Afrad Basheer, and Sandeep K. Goyal. *Quantum k-nearest neighbor machine learning algorithm.* Tech. rep. 2020. arXiv: 2003.09187. URL: http://arxiv.org/abs/2003.09187.

[122] Ana Martin et al. *Towards Pricing Financial Derivatives with an IBM Quantum Computer.* Tech. rep. 2019. arXiv: 1904.05803. URL: http://arxiv.org/abs/1904.05803.

[123] Juan Miguel Arrazola et al. *Quantum-inspired algorithms in practice.* Tech. rep. 2019. arXiv: 1905.10415. URL: http://arxiv.org/abs/1905.10415.

[124] Dawid Kopczyk. *Quantum machine learning for data scientists.* Tech. rep. 2018. arXiv: 1804.10068. URL: http://arxiv.org/abs/1804.10068.

[125] Adrián Pérez-Salinas et al. "Data re-uploading for a universal quantum classifier". In: *Quantum* 4 (July 2020), p. 226. ISSN: 2521327X. DOI: 10.22331/q-2020-02-06-226. arXiv: 1907.02085. URL: http://arxiv.org/abs/1907.02085%20http://dx.doi.org/10.22331/q-2020-02-06-226.

[126] Vojtěch Havlíček et al. *Supervised learning with quantum-enhanced feature spaces.* Tech. rep. 7747. 2019, pp. 209–212. DOI: 10.1038/s41586-019-0980-2. arXiv: 1804.11326.

[127] Nana Liu and Patrick Rebentrost. *Quantum machine learning for quantum anomaly detection.* Tech. rep. 4. 2018. DOI: 10.1103/PhysRevA.97.042315. arXiv: 1710.07405.

[128] Iris Cong and Luming Duan. “Quantum discriminant analysis for dimensionality reduction and classification”. In: *New Journal of Physics* 18.7 (2016). ISSN: 13672630. DOI: 10.1088/1367-2630/18/7/073011. arXiv: 1510.00113.

[129] Chen Ding, Tian-Yi Bao, and He-Liang Huang. *Quantum-Inspired Support Vector Machine*. Tech. rep. 8. 2019. arXiv: 1906.08902. URL: http://arxiv.org/abs/1906.08902.

[130] Mateusz Ostaszewski, Piotr Gawron, and Przemysław Sadowski. *Quantum image classification using principal component analysis*. Tech. rep. 1. 2015, pp. 1–12. DOI: 10.20904/271001. arXiv: 1504.00580.

[131] Engin Şahín and Ihsan Yilmaz. “QRMW: Quantum representation of multi wavelength images”. In: *Turkish Journal of Electrical Engineering and Computer Sciences* 26.2 (2018), pp. 768–779. ISSN: 13036203. DOI: 10.3906/elk-1705-396.

[132] Yi Zhang et al. “NEQR: A novel enhanced quantum representation of digital images”. In: *Quantum Information Processing* 12.8 (2013), pp. 2833–2860. ISSN: 15700755. DOI: 10.1007/s11128-013-0567-z. URL: https://www.researchgate.net/publication/257641933.

[133] Hai Sheng Li et al. “A Quantum Image Representation Based on Bitplanes”. In: *IEEE Access* 6 (2018), pp. 62396–62404. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2871691.

[134] Phuc Q. Le, Fangyan Dong, and Kaoru Hirota. “A flexible representation of quantum images for polynomial preparation, image compression, and processing operations”. In: *Quantum Information Processing* 10.1 (2011), pp. 63–84. ISSN: 15700755. DOI: 10.1007/s11128-010-0177-y.

[135] Ville Bergholm et al. “PennyLane: Automatic differentiation of hybrid quantum classical computations”. In: *arXiv* (Nov. 2018). ISSN: 23318422. arXiv: 1811.04968. URL: http://arxiv.org/abs/1811.04968.

[136] Abraham Asfaw et al. *Learn Quantum Computation Using Qiskit*. 2020. URL: http://community.qiskit.org/textbook.

[137] Ni Kadek Ayu Wirdiani et al. “Face Identification Based on K-Nearest Neighbor”. In: *Scientific Journal of Informatics* 6.2 (2019), pp. 150–159. ISSN: 2407-7658. DOI: 10.15294/sji.v6i2.19503. URL: http://journal.unnes.ac.id/nju/index.php/sji.

[138] Sandeep Kumar, Sukhwinder Singh, and Jagdish Kumar. “Multiple face detection using hybrid features with SVM classifier”. In: *Advances in Intelligent Systems and Computing* 847 (2019), pp. 253–265. ISSN: 21945357. DOI: 10.1007/978-981-13-2254-9_23. URL:

https://link.springer.com/chapter/10.1007/978-981-13-2254-9%7B%5C_%7D23.

[139] Jin Min Liang et al. *Variational quantum algorithms for dimensionality reduction and classification.* Tech. rep. 3. 2020. DOI: 10.1103/PhysRevA.101.032323. arXiv: 1910.12164.

[140] Yong Liu et al. *Variational quantum circuits for quantum state tomography.* Tech. rep. 5. 2020. DOI: 10.1103/PhysRevA.101.052316. arXiv: 1912.07286.

[141] Johannes Bausch. *Recurrent Quantum Neural Networks.* Tech. rep. 2020. arXiv: 2006.14619v1.

[142] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. *Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms.* Tech. rep. 12. 2019, p. 1900070. DOI: 10.1002/qute.201900070. arXiv: 1905.10876.

[143] Thomas Hubregtsen et al. *Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability.* Tech. rep. 1. 2021. DOI: 10.1007/s42484-021-00038-w. arXiv: 2003.09887. URL: https://www.researchgate.net/publication/340115185.

[144] William Cappelletti, Rebecca Erbanni, and Joaquín Keller. *Polyadic Quantum Classifier.* Tech. rep. 2020. arXiv: 2007.14044. URL: http://arxiv.org/abs/2007.14044.

[145] Adriano Barenco et al. *Elementary gates for quantum computation.* Tech. rep. 5. 1995, pp. 3457–3467. DOI: 10.1103/PhysRevA.52.3457. arXiv: 9503016 [quant-ph].

[146] Jarrod R. McClean et al. "Barren plateaus in quantum neural network training landscapes". In: *Nature Communications* 9.1 (2018). ISSN: 20411723. DOI: 10.1038/s41467-018-07090-4. arXiv: 1803.11173. URL: www.nature.com/naturecommunications.

[147] M. Cerezo et al. *Cost function dependent barren plateaus in shallow parametrized quantum circuits.* Tech. rep. 1. 2021. DOI: 10.1038/s41467-021-21728-w. arXiv: 2001.00550.

[148] Andrea Skolik et al. *Layerwise learning for quantum neural networks.* Tech. rep. 1. 2021. DOI: 10.1007/s42484-020-00036-4. arXiv: 2006.14904.

[149] Edward Grant et al. *An initialization strategy for addressing barren plateaus in parametrized quantum circuits.* Tech. rep. 2019. DOI: 10.22331/q-2019-12-09-214. arXiv: 1903.05076.

[150] James Stokes et al. *Quantum Natural Gradient.* Tech. rep. 2020. DOI: 10.22331/Q-2020-05-25-269. arXiv: 1909.02108.

[151] Naoki Yamamoto. *On the natural gradient for variational quantum eigensolver.* Tech. rep. 2019. arXiv: 1909.05074. URL: http://arxiv.org/abs/1909.05074.

[152] Yasunari Suzuki et al. *Qulacs: a fast and versatile quantum circuit simulator for research purpose.* Tech. rep. 2020. arXiv: 2011.13524. URL: http://arxiv.org/abs/2011.13524.

[153] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[154] Ryan LaRose and Brian Coyle. *Robust data encodings for quantum classifiers.* Tech. rep. 2020. arXiv: 2003.01695. URL: http://arxiv.org/abs/2003.01695.

[155] Carlo Ciliberto et al. "Quantum machine learning: A classical perspective". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2209 (2018). ISSN: 14712946. DOI: 10.1098/rspa.2017.0551. arXiv: 1707.08561. URL: http://dx.doi.org/10.1098/rspa.2017.0551.

[156] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.* International Conference on Learning Representations, ICLR, Dec. 2015. arXiv: 1412.6980. URL: https://arxiv.org/abs/1412.6980v9.

[157] Laith Alzubaidi et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *J Big Data* 8 (2021), p. 53. DOI: 10.1186/s40537-021-00444-8. URL: https://doi.org/10.1186/s40537-021-00444-8.

[158] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017), pp. 84–90. ISSN: 15577317. DOI: 10.1145/3065386. URL: http://code.google.com/p/cuda-convnet/.

[159] Guoqing Li et al. "Efficient densely connected convolutional neural networks". In: *Pattern Recognition* 109 (Jan. 2021), p. 107610. ISSN: 00313203. DOI: 10.1016/j.patcog.2020.107610.

[160] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Sept. 2014). arXiv: 1409.1556. URL: https://arxiv.org/abs/1409.1556v6.

[161] Saining Xie et al. "Aggregated Residual Transformations for Deep Neural Networks". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January (Nov. 2016), pp. 5987–5995. arXiv: 1611.05431. URL: https://arxiv.org/abs/1611.05431v2.

[162] Joseph Redmon and Ali Farhadi. *YOLOv3: An Incremental Improvement.* Apr. 2018. arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

[163] Haenssle HA et al. "Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists". In: *Annals of oncology : official journal of the European Society for Medical Oncology* 29.8 (Aug. 2018), pp. 1836–1842. ISSN: 1569-8041. DOI: 10.1093/ANNONC/MDY166. URL: https://pubmed.ncbi.nlm.nih.gov/29846502/.

[164] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (Jan. 2017), pp. 115–118. ISSN: 14764687. DOI: 10.1038/nature21056. URL: https://www.nature.com/articles/nature21056.

[165] Cicero M et al. "Training and Validating a Deep Convolutional Neural Network for Computer-Aided Detection and Classification of Abnormalities on Frontal Chest Radiographs". In: *Investigative radiology* 52.5 (May 2017), pp. 281–287. ISSN: 1536-0210. DOI: 10.1097/RLI.0000000000000341. URL: https://pubmed.ncbi.nlm.nih.gov/27922974/.

[166] Cheng JZ et al. "Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans". In: *Scientific reports* 6 (Apr. 2016). ISSN: 2045-2322. DOI: 10.1038/SREP24454. URL: https://pubmed.ncbi.nlm.nih.gov/27079888/.

[167] Bao Wang et al. "Deep Learning for Real Time Crime Forecasting". In: (July 2017). arXiv: 1707.03340. URL: https://arxiv.org/abs/1707.03340v1.

[168] Shashi Yadav. "Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence". In: *International Journal for Research in Applied Science and Engineering Technology* 8.7 (July 2020), pp. 1368–1375. DOI: 10.22214/IJRASET.2020.30560.

[169] Manjia Wu et al. “Real-time drone detection using deep learning approach”. In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST* 251 (2018), pp. 22–32. DOI: 10.1007/978-3-030-00557-3_3.

[170] Ismini Psychoula et al. “A Deep Learning Approach for Privacy Preservation in Assisted Living”. In: *2018 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018* (Feb. 2018), pp. 710–715. arXiv: 1802.09359. URL: https://arxiv.org/abs/1802.09359v1.

[171] Maria Ahmed Qureshi et al. “Deep learning-based ambient assisted living for self-management of cardiovascular conditions”. In: *Neural Computing and Applications 2021* (Jan. 2021), pp. 1–19. ISSN: 1433-3058. DOI: 10.1007/S00521-020-05678-W. URL: https://link.springer.com/article/10.1007/s00521-020-05678-w.

[172] Erinc Merdivan. “Ambient Assisted Living with Deep Learning”. PhD thesis. CentraleSupélec, 2019.

[173] Ugnius Malūkas et al. “Real time path finding for assisted living using deep learning”. In: *Journal of Universal Computer Science* 24.4 (2018), pp. 475–487. ISSN: 09486968.

[174] Philip Easom-Mccaldin et al. “On Depth, Robustness and Performance Using the Data Re-Uploading Single-Qubit Classifier”. In: *IEEE Access* 9 (2021), pp. 65127–65139. DOI: 10.1109/ACCESS.2021.3075492.

[175] Jacob Biamonte et al. *Quantum machine learning.* Tech. rep. 7671. 2017, pp. 195–202. DOI: 10.1038/nature23474. arXiv: 1611.09347.

[176] Li Deng. “The MNIST database of handwritten digit images for machine learning research”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142. ISSN: 10535888. DOI: 10.1109/MSP.2012.2211477.

[177] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: *arXiv* (Aug. 2017). arXiv: 1708.07747. URL: https://arxiv.org/abs/1708.07747v2.

[178] F. S. Samaria and A. C. Harter. “Parameterisation of a stochastic model for human face identification”. In: *IEEE Workshop on Applications of Computer Vision - Proceedings* (1994), pp. 138–142. DOI: 10.1109/acv.1994.341300.

[179] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[180] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. “Effect of data encoding on the expressive power of variational quantum-machine-learning models”. In: *Physical Review A* 103.3 (2021). ISSN: 24699934. DOI: `10.1103/PhysRevA.103.032430`. arXiv: `2008.08605`.