

# Efficient Quantum Image Classification Using Single Qubit Encoding

Philip Easom-McCaldin<sup>ID</sup>, Ahmed Bouridane<sup>ID</sup>, Senior Member, IEEE, Ammar Belatreche<sup>ID</sup>, Member, IEEE,  
Richard Jiang<sup>ID</sup>, and Somaya Al-Maadeed<sup>ID</sup>, Senior Member, IEEE

**Abstract**—The domain of image classification has been seen to be dominated by high-performing deep-learning (DL) architectures. However, the success of this field, as seen over the past decade, has resulted in the complexity of modern methodologies scaling exponentially, commonly requiring millions of parameters. Quantum computing (QC) is an active area of research aimed toward greatly reducing problems of complexity faced in classical computing. With growing interest toward quantum machine learning (QML) for applications of image classification, many proposed algorithms require usage of numerous qubits. In the noisy intermediate-scale quantum (NISQ) era, these circuits may not always be feasible to execute effectively; therefore, we should aim to use each qubit as effectively and efficiently as possible, before adding additional qubits. This article proposes a new single-qubit-based deep quantum neural network for image classification that mimics traditional convolutional neural network (CNN) techniques, resulting in a reduced number of parameters compared with previous works. Our aim is to prove the concept of the initial proposal by demonstrating classification performance of the single-qubit-based architecture, as well as to provide a tested foundation for further development. To demonstrate this, our experiments were conducted using various datasets including MNIST, Fashion-MNIST, and ORL face datasets. To further our proposal in the context of the NISQ era, our experiments were intentionally conducted in noisy simulation environments. Initial test results appear promising, with classification accuracies of 94.6%, 89.5%, and 82.5% achieved on the subsets of MNIST, FMNIST, and ORL face datasets, respectively. In addition, proposals for further investigation and development were considered, where it is hoped that these initial results can be improved.

**Index Terms**—Quantum convolutional neural networks (CNNs), quantum deep learning (DL), quantum facial biometrics, single-qubit encoding.

Manuscript received 1 October 2021; revised 25 April 2022; accepted 20 May 2022. Date of publication 17 June 2022; date of current version 6 February 2024. This work was supported in part by the Qatar National Research Fund (a member of the Qatar Foundation) under Grant NPRP11S-0113-180276 and in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/P009727/1. (*Corresponding author: Philip Easom-McCaldin.*)

Philip Easom-McCaldin and Ammar Belatreche are with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: philip.easom@northumbria.ac.uk; ammar.belatreche@northumbria.ac.uk).

Ahmed Bouridane is with the Centre for Data Analytics and Cybersecurity (CDAC), University of Sharjah, Sharjah, United Arab Emirates (e-mail: abouridane@sharjah.ac.ae).

Richard Jiang is with the Department of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K. (e-mail: r.jiang2@lancaster.ac.uk).

Somaya Al-Maadeed is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar (e-mail: s.alali@qu.edu.qa).

Digital Object Identifier 10.1109/TNNLS.2022.3179354

## I. INTRODUCTION

IMAGE classification has seen rapid improvements over the past decade alone. The processing capability of readily available GPU units has enabled a chain of strong-performing deep-learning (DL) methodologies [1]–[8] to dominate the field, boasting high levels of classification accuracy that can be fine-tuned to a specific task. The result of this is that machine learning (ML) has been able to become integrated within society for many social and industrial uses, e.g., healthcare [9]–[12], public safety [13]–[15], and assisted living [16]–[19].

While 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. As an example of this growth of complexity within DL, one of the earliest convolutional neural network (CNN) methods, LeNet5 [20], has a total of ~60 000 parameters and can reach test-set accuracy values over 98%. In contrast to this, one of the top-performing methods [21] reaching an accuracy value of 99.83% requires a mere 1 400 000 parameters, over 23× that of LeNet5 for 1%–2% increase in accuracy performance.

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 toward 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 DL has shown that, by focusing on the development of methods that have a more efficient usage of parameters, a foundation can be provided to build upon and progress toward the highest performance levels of classification while keeping efficiency of training and execution a primary factor.

Quantum computing (QC) has undergone a tremendous level of development within the past few years, with quantum machine learning (QML) seeing a large increase in attention and productivity. Through innate parallelism and fast execution speeds, it is supported by many that QC may provide the necessary means to overcome classification performance plateaus seen throughout classical ML, and ultimately progress toward effective, yet efficient ML algorithms. Even though QC is in its infancy, or noisy, intermediate-scale quantum (NISQ) stage, progress has been made toward the development of standalone QML algorithms that are capable classifiers in themselves [22], [24], [28].

In this work, we aim to progress the thorough work conducted toward 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, our initial experiments show-

case results conducted on lower level image-classification tasks, following the natural progression of dataset complexity. These experiments are also conducted using noisy and nonnoisy simulation environments, in order to provide a reasonable expectation of how the method will perform in the current NISQ era.

The findings from our results show that as low as six parameters are enough to form a suitably complex feature space, capable of classifying image data to a high degree of accuracy. Alongside this, experimental results show a factor of robustness against the phase damping noise channel to some extent.

The concept of single-qubit-based neural networks has been presented by Pérez-Salinas *et al.* [36] and is analogous to a simple multilayer perceptron (MLP), with only one dense hidden layer and tests on several toy datasets. In our work, we aim to expand this strategy to quantum image classification and develop new architectures such as quantum CNNs, which are often considered as a much more complex structure than a simple MLP [54].

To bring the single-qubit strategy into quantum CNNs, we propose several methods to implement our new single-qubit quantum CNNs. Particularly: 1) we design a method that maintains spatial relationships of pixels through the use of parametrized convolutional filters and 2) we adapt this method to process images in their natural form, thus not requiring a costly image flattening preprocessing step. Consequently, we can then easily implement the quantum CNNs via single-qubit-based data uploading.

When considering the contributions of this work, it is also important to consider the indirect contributions that arise from the modifications made. First, the proposed method has an increased specificity to the domain of quantum image classification in comparison with prior work shown in [36]. The proposed framework also enables modular-based architectures to be developed using QML techniques, therefore allowing significant room for further expansion and development. Furthermore, we extended our work to an emerging topic, namely quantum biometrics, and successfully tested our proposed new single-qubit quantum CNNs on facial biometrics besides the handwriting dataset to a promising extent.

Overall, the work presented here is an important step that expands upon a single-qubit encoding approach toward a more practical, long-term solution that is not only more adaptable in nature but also more efficient when scaled to larger dimensions.

The structure of this article is organized as follows. First, related work in the field of QML is discussed, and derivation of the proposed method via single-qubit encoding principles is outlined. Then, the experimental setup is described in relevance to the current capabilities of QML classifiers. Afterward, our experimental results are shown, 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.

## II. RELATED WORK

In its current state, many NISQ QML algorithms tend to use a backbone of variational quantum circuits (VQC) as their primary computational tool. VQCs typically consist of a series of single-qubit and multiqubit unitary gate operations

applied using a set of parameters in a linear fashion over a number of qubits [22]–[28]. Some of these VQC algorithms are presented as a hybrid approach to computation, working in conjunction with typical classical processes implemented as pre- or postprocessing to determine a classification result.

Hybrid approaches of computation may provide an opportunity to utilize the power of QC with predetermined methods, e.g., classically extracted features fed through a QVC [29], or vice versa with quantum-extracted features [30]. Experimental results within [30] suggest that quantum-extracted features may provide a small advantage to classification performance over a purely classical framework. However, it was difficult to distinguish between a third method with randomly implemented nonlinearities; therefore, it may not always be clear to unequivocally identify the impact that quantum processes have on classification results.

Within classical ML, DL CNN algorithms are typically employed for image classification tasks. CNN algorithms traditionally implement convolutional, nonlinear transformation and pooling operations as a series of layers, prior to a fully connected portion to determine a classification result. Motivated once more by the success of CNN methods, recent works have proposed fully quantum architectures as similarly based alternatives. Work presented in [31] mimics the traditional convolutional-pooling layer series through the application of successive multiqubit unitary operations followed by qubit measurement. Here, nonlinearity is introduced by utilizing the measurement result of particular qubits as rotational parameters.

In separate work, Kerenidis *et al.* [32] propose a quantum CNN that computes the forward pass of the algorithm via quantum inner product estimation between an input and convolutional kernel. Then, nonlinearity is introduced via a Boolean circuit function. Rotational operations and amplitude amplification are then performed to enable pixels of a higher value to have a higher measurement probability. Individual experiments for the method of [31] and [32] have shown promising results for image classification using MNIST data, as well as for a quantum error correction task. However, these other methods discussed rely on the entire input data to be encoded in the amplitudes of a many-qubit superposition state, i.e., amplitude encoding.

While the work discussed throughout this section has had promising results and shows positivity toward the development of effective quantum classifiers using many qubits, it is important to remain in context with the current NISQ era of quantum computation. Therefore, we should 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, multiqubit operations, where a small error could vastly impact the states of other qubits utilized. By developing toward small-scale, efficient methods using minimal qubits, a solid foundation can be built to progress in the quest for effective QML image classification algorithms.

In an effort to find efficient, yet effective data encoding schemes, recent works [33]–[35] have analyzed a variety of QVC structures to determine the ability of the encoding to navigate the Bloch sphere (referred to as expressability), capability of entanglement between qubits, as well as robustness when realized in a noisy quantum environment. Within these works, it was identified that there was a strong correlation between expressability and classification accuracy. However,

it was noted that a point of saturation exists for expressability, as circuit depth is increased.

One method in particular [36] remained consistent with these findings, while additionally seeing promising results as a capable classifier. This method encoded the input vector as a set of weighted parameters over a series of arbitrary unitary operations. Varied settings for depth could then be initialized, where an increased depth did show a correlation for improved performance on par with classical neural networks and support-vector machine classifiers. What makes this proposal particularly appealing is its capability to encode an arbitrary amount of data into a complex feature space, while requiring the use of a single qubit only. The proposal of this work was examined further in [37], 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 toward 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 forward in an effort to create effective, robust quantum classifiers, similar to the rise of state-of-the-art DL methodologies dominating many classical ML problems.

### III. METHODOLOGY

#### A. 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 are 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).

While these data encoding schemes have been employed within other works or [53]–[55], 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 the usage of qubits.

Single-qubit encoding, developed in [36], 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  $2 \times 2$  unitary matrix of determinant 1), the corresponding operation is able to be decomposed into the following three rotational operations [38]:

$$U = e^{i\alpha} R_Z(\beta)R_Y(\gamma)R_Z(\delta). \quad (1)$$

With a global phase factor  $\alpha$ , Euler angles  $\beta, \gamma, \delta \in \mathbb{R}$  that define the extent of each rotation ( $R$ ) around the  $Z$ -,  $Y$ -, and  $Z$ -axes, respectively. It is noted that the unitary operation does not require an  $R_x$  rotation. Within this method of encoding, these Euler angles are parameterized further and defined as

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

where  $\theta_i$  and  $\phi_i$  are trainable weight parameters assigned to  $x_i$ , the value of the input vector  $x$  at dimension  $i$ . Therefore, the extent of rotation  $\beta, \gamma, \delta$  is with respect to the weighted value of the input. 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 3-D 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  $\theta$  and  $\phi$  variable. Therefore, for this example, a total of 288 parameters are required to encode the information fully.

#### B. 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  $F \times F$  pixels. In classical ML, a kernel operation would be applied to result in a value for that region of pixels.

The first step in our proposed modification is to adopt a similar approach to this. Rather than flatten an image into the form of a column vector as a preprocessing step, the original shape of the image is maintained. A filter of size  $F \times F$  is then passed over the image, partitioning the image into a distinct grid of  $F \times F$  squares. Each square region of pixels is then encoded onto the qubit in turn row by row using the described single-qubit encoding scheme with pixel values ( $x_i$ ) and respective filter weights as parameters ( $\theta_i, \phi_i$ ).

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  $F \times F$  region of pixels, a set of six weight parameters are assigned to the filter itself, which correspond to  $\theta$  and  $\phi$  in (2). By doing so, the same set of six parameters will repeatedly be applied to every series of three pixels that the filter has extracted. This method reduces the number of parameters required to just six per filter.

While it is acknowledged that multiple unique sets of six parameters could be localized to each  $F \times F$  region, our aim is to demonstrate that it is possible to produce reasonable results with the fewest parameters. Therefore, all experiments contained within this work will be conducted using a system setup of a single filter with six parameters in total, as displayed in Fig. 1. However, both setups discussed offer a slightly different approach toward 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. More considerations toward possible future work and extensions will be included later in the discussion section of this article.

#### C. Classification Pipeline and Loss Calculation

So far, the proposed encoding strategy has been defined in Section III-B; however, the flow from input to classification

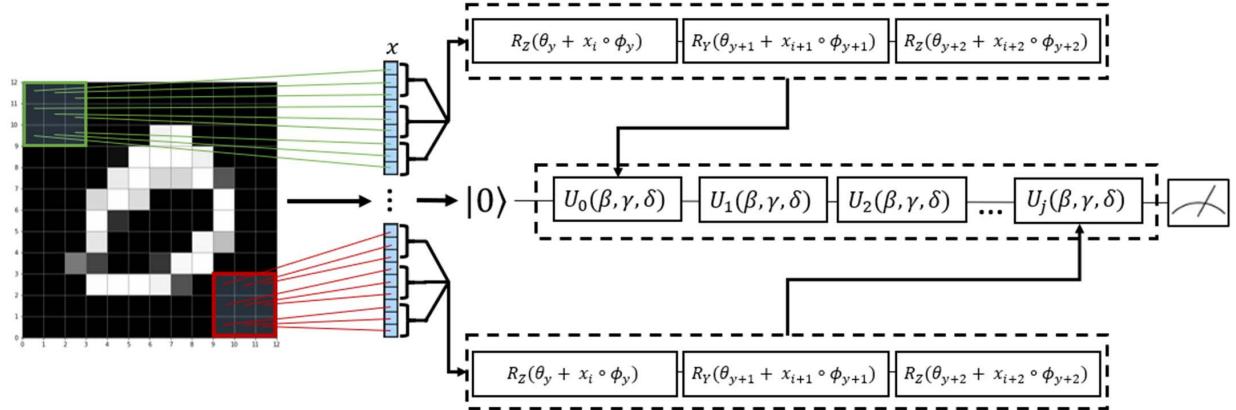


Fig. 1. Overview of the proposed methodology with both modifications made. First, a filter is applied over an image ( $12 \times 12$  shown), where square regions of pixels are extracted. These regions are flattened in turn to form a column vector and encoded using the single-qubit encoding scheme, cycling through the six weights contained in  $\theta$  and  $\phi$ . This process is repeated until the filter has processed over the entire image, where measurement is taken with respect to a target state given. The number of square pixel regions to encode and the number of unitary operations required  $j$  are determined by the size of the filter  $F \times F$ , as well as the stride value  $S$  used.

output has not been made evident. To do this, a fidelity-based approach of measurement is adopted as seen in [36], 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 fidelity 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 [36]:

$$\frac{1}{2D} \sum_{m=1}^D \sum_{c=1}^C ((F(x_d, \theta, \phi)_c - F_c)^2) \quad (3)$$

where  $D$  is the set size of images used,  $C$  is the number of classes,  $F(x_d, \theta, \phi)_c$  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  $|\psi\rangle = (|0\rangle + |1\rangle)/(\sqrt{2})^{1/2}$ , then the fidelity measurement is given by

$$F(x_d, \theta, \phi)_c = |\langle \psi_c | \psi(x_d, \theta, \phi) \rangle|^2. \quad (4)$$

Here,  $F(x_d, \theta, \phi)_c = 0.5$  for  $c = 0$ . Expected fidelity values can also be found using (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 hardcoded variables in lines 12 and 18, the value in line 12 relates to the three required values per unitary gate  $\beta$ ,  $\gamma$ , and  $\delta$ , 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 18 refers to the successive application of unitary operations, where the cycling of  $i$  in multiples of 3 allows for the three unitary operation values  $\beta$ ,  $\gamma$ , and  $\delta$  to be given in turn.

#### IV. RESULTS

In this section, our experiments conducted using the methodology described throughout Section III will be presented, where an initial analysis will be conducted into the results obtained. Our first experiments will be conducted using a subset of the MNIST data, used for both binary and three-class classification tasks. The MNIST dataset [39] 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 [40] for binary and three-class classification tasks. FMNIST data are often considered a subsequently more challenging task than MNIST data, so it poses an appropriate challenge for the low-parameter system to tackle effectively. Third, the methodology will be applied to a face identification task, using a custom dataset consisting of AT&T face images [41], as well as a collection of images taken at random from the CIFAR10 dataset [42]. For all sets of experiments, the classes and index values of data used remained consistent. This ensured that the experimental results obtained could be compared in a fair and justifiable manner.

Finally, it is important to consider the impact that environmental noise has on the capability of the algorithm presented.

**Algorithm 1** Proposed Methodology

**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  $Q$

```

1: For image  $d \in D$ : (Image of height  $H$  and width  $W$ )
2:   For filter  $f \in N_f$ : (Filter of size  $F \times F$  and stride  $S$ )
3:     While pixel row  $d_{row} < H$ : (Cycle filter regions)
4:       While pixel column  $d_{col} < W$ :
5:         For filter row  $f_{row} = 1, \dots, F$ : (Gather pixels)
6:           For filter column  $f_{col} = 1, \dots, F$ :
7:              $r = (d_{row} + f_{row}); c = (d_{col} + f_{col})$ 
8:             If  $r < H$  and  $c < W$ :
9:                $x$  append value at pixel  $(r, c) \in d$ 
10:            Else:
11:               $x$  append 0
12:            If  $\text{len}(x) \% 3 != 0$ : (% = modulo operation)
13:               $x$  append 0
14:             $d_{col} \rightarrow d_{col} + S$ 
15:             $d_{row} \rightarrow d_{row} + S$ 
16: For filter  $f \in N_f$ : (Quantum circuit application)
17:   For  $\psi_c \in C$ : (Cycle class states)
18:     For  $i = 1 : 3 : x_{max}$ : (Apply weights in sets of 3)
19:        $\beta = \theta_0 + x_i \cdot \phi_0$ 
20:        $\gamma = \theta_1 + x_{i+1} \cdot \phi_1$ 
21:        $\delta = \theta_2 + x_{i+2} \cdot \phi_2$ 
22:       Apply  $U(\beta, \gamma, \delta)$ 
23:     Measure qubit fidelity  $F(x_d, \theta, \phi)_c$  w.r.t  $\psi_c$ 
24:      $Q_d$  append  $F(x_d, \theta, \phi)_c$ 
25: End

```

Given that our experiments are conducted in a simulation environment, our noise implementation will also be simulated but it makes appropriate use of various noise and distortion channels to produce realistic and effective results.

To provide general details of the experimental setup and implementation, the framework for these experiments was developed using the PennyLane library [43], which also incorporated usage of the PyTorch interface [44]. For nonnoisy simulations, the Qulacs [45] 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 pseudorandom number generation seeds were set to zero.

For initialization, all weights were formed using a Gaussian distribution with a mean of 0 and a 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 [46], it is relevant to address the consideration taken toward this. While it is acknowledged that there have been some proposals toward overcoming the problem of barren plateaus, namely through localized cost functions [47], usage of quantum natural gradients [48], and evaluations of initial weight selections [49], we did not incorporate any specific approaches to reduce their occurrence. 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;

TABLE I  
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

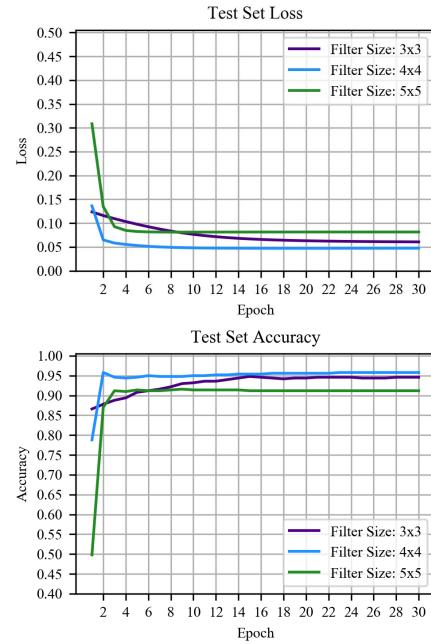


Fig. 2. Test set loss and accuracy results of a binary classification task, conducted with a single filter of varied size on a subset of the MNIST data with classes 0 and 1.

however, there is no common practice as of yet to overcome this problem to our knowledge; therefore, it was felt that our course of action was appropriate for now.

**A. 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 per class used. For each experiment, 30 epochs of optimization were conducted using the Adam [50] 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. Thirty epochs of optimization were also selected from initial experiments, as satisfactory convergence could be reached within the time-frame, while not requiring extremely long training periods.

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

Upon inspection of the test set loss and accuracy curves displayed in Fig. 2, it can be seen that the behavior of the curve for the  $3 \times 3$  filter is different from that of  $4 \times 4$  and  $5 \times 5$ . Here, the curve for the  $3 \times 3$  filter experiment begins

TABLE II

FINAL CLASSIFICATION PERFORMANCE VALUES AT EPOCH 30 FOR VARIOUS FILTER SIZES USING MULTICLASS 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

at a more favorable standard of classification performance, but any improvements occur slowly and gradually over the course of training.

The latter experiments of filter sizes  $4 \times 4$  and  $5 \times 5$  begin in a more unfavorable position with lower loss and accuracy values; however, the initial improvements to classification performance are very sharp and quickly plateau by approximately epoch 5. The behavior exhibited here suggests that while the initial weight distribution for the  $3 \times 3$  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  $3 \times 3$  will overtake the  $4 \times 4$  filter experiment. However, the  $3 \times 3$  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 toward 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 five epochs.

Table II shows final performance values taken at epoch 30 from experiments conducted on multiclass (three-class) MNIST data using classes 0, 1, and 2 and a varied filter size. Within this, a filter size of  $3 \times 3$  produced the best classification performance overall, followed by the  $5 \times 5$  filter and  $4 \times 4$  filter, respectively.

While the results achieved here may not be state of the art, there are some points which must be considered in context of this work. 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 and their target class state, this naturally becomes more difficult to achieve with an increased number of classes, provided 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 Fig. 3 show, this lower embedding complexity thus equates to a plateau, or extremely marginal improvements in both loss and accuracy over time.

In order to demonstrate this, Fig. 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

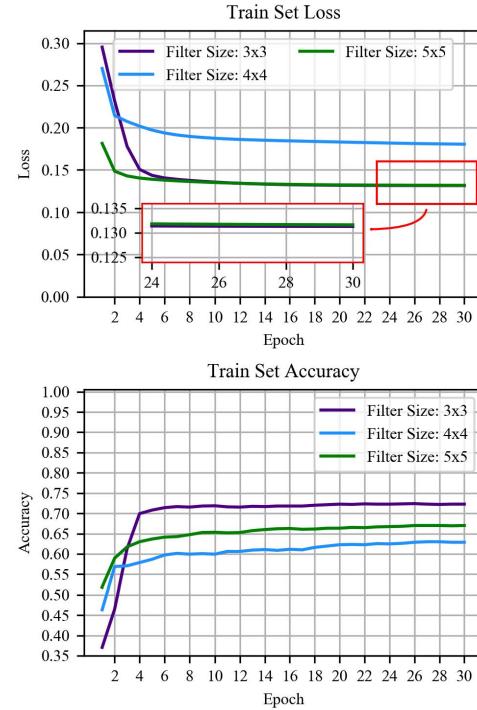


Fig. 3. Train set loss and accuracy results of the experiment using three-class (classes 0, 1, and 2) MNIST data with varied filter sizes. For clarity, the inset box within the train set loss chart displays the  $3 \times 3$  filter line just below the  $5 \times 5$  filter line.

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  $5 \times 5$  filter produced a much denser embedding of all datapoints in this case. In contrast, embeddings from  $3 \times 3$  and  $4 \times 4$  filters were fairly similar in that the datapoints are more widely distributed toward their respective target states overall, with the  $3 \times 3$  filter experiment arguably showing the most distinctive distributions of datapoints per class. However, despite these differences, the loss value of the  $3 \times 3$  experiment is very slightly below the  $5 \times 5$  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 color groups of datapoints for the  $5 \times 5$  experiment. 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 given previously (equating to various image classes). However, there is a large section toward 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  $3 \times 3$  filter experiment, where the datapoint clusters were spread more widely yet remained fairly compact.

While the  $4 \times 4$  filter experiment produced the poorest classification performance results overall, the resulting embeddings show that this experiment struggled to form a significant class cluster consisting of the yellow datapoints, and so had many incorrectly classified images as a result. Had the  $4 \times 4$  filter experiment been more successful in doing this, then it could be argued that the final embeddings of the image

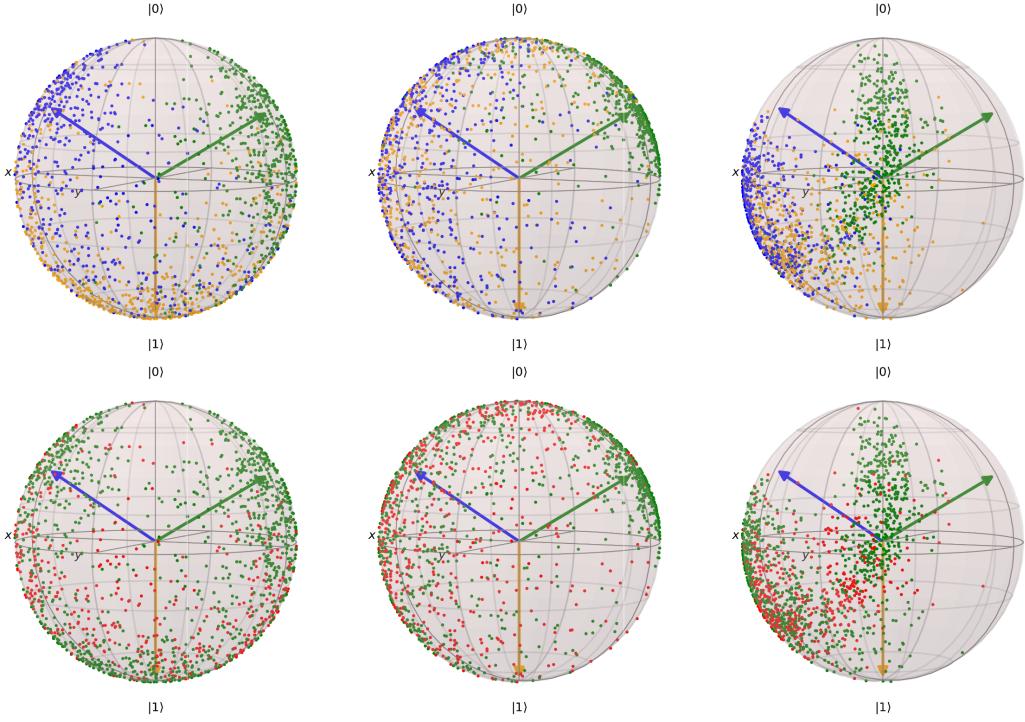


Fig. 4. Visualizations of Bloch sphere embeddings of datapoints corresponding to dataset images for the experiment conducted using three-class MNIST data. Left to right: Bloch spheres show train set data embeddings taken over epoch 30 for the  $3 \times 3$ ,  $4 \times 4$ , and  $5 \times 5$  filter size experiments, respectively. For the top row, point colors correspond to the images' respective class, whereas for the bottom row, green points represent correctly classified datapoints, and red points represent incorrectly classified datapoints. For all Bloch spheres, the three central arrows represent the target state vector for that color class.

TABLE III  
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	0.12952	0.824	0.16690	0.790
5x5	0.17981	0.726	0.25194	0.575

data would act similar to the  $3 \times 3$  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 six 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.

### B. 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. 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 III displays classification performance values from experiments on binary FMNIST data, using classes 0 (t-shirt) and 1 (trousers) with varied filter sizes. From these results, a filter size of  $3 \times 3$  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 increases.

When inspecting the train set and test set loss curves displayed in Fig. 5, the behavior of all three experiments appears to contradict one another to some extent. While the  $3 \times 3$  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  $4 \times 4$  experiment shows a slow and gradual decrease in loss, and the  $5 \times 5$  experiment displays very little convergence and plateaus close to the initial loss value after epoch 1.

The charts displayed in Fig. 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 effectively to 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  $4 \times 4$  filter experiment could be in an area of lower gradient within the loss landscape, resulting in the behavior displayed and described earlier being drawn out over a longer period of time.

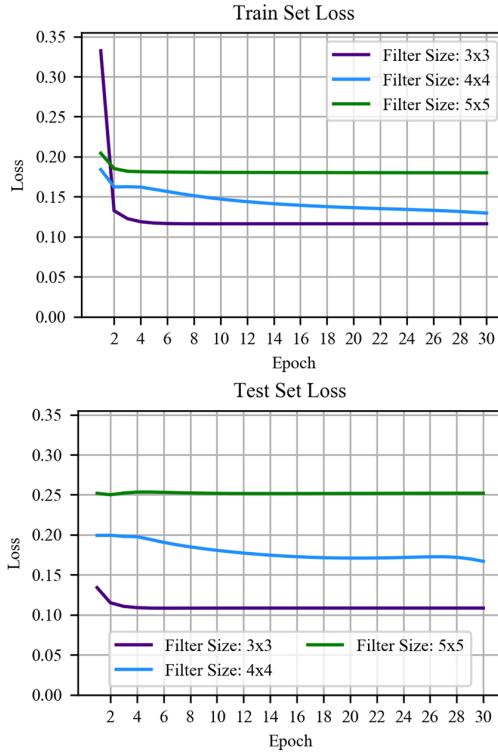


Fig. 5. Train set and test set loss curves relating to experimental results displayed in Table III. The experimental data consisted of a subset of the FMNIST dataset using image classes of 0 and 1.

In order to visualize this, Fig. 6 displays various Bloch sphere embeddings of train set data at epochs 1, 2, and 30 with point color corresponding to class value. For the left-hand Bloch sphere plot at epoch 1, the distribution of datapoints is fairly dense toward the 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 training epoch, the second set of embeddings for epoch 2 is more evenly distributed between the two hemispheres. Visually, as the datapoints are embedded closer to 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 and embed them closer toward 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 that 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 toward 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

TABLE IV  
FINAL CLASSIFICATION PERFORMANCE VALUES AT EPOCH 30  
USING VARIOUS FILTER SIZES FOR CUSTOM FACIAL  
IDENTIFICATION DATASET

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

within this experiment, a test set classification accuracy close to 90% was achieved. In the context of this work, 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 work and analysis.

### C. Facial Identification and Facial Recognition Results

Section IV-C 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 nonface (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 with all experiments, hyperparameter values were selected via 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 IV. As with previous multiclass MNIST and FMNIST experimental results, the filter size of  $3 \times 3$  produces the highest performance overall. When viewing the graph of training set loss, displayed in the top half of Fig. 7, the loss values for  $4 \times 4$  and  $5 \times 5$  filter size experiments are very similar, and appear to plateau at the same epoch. However, the loss curve for the  $3 \times 3$  filter experiment does not appear to plateau in this experiment over the number of optimization epochs conducted.

By visualizing the associated image embeddings in the bottom half of Fig. 7, 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 case of binary classification). As optimization continues by epoch 30, it can be seen by the right-hand Bloch sphere that while 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 throughout training for the  $3 \times 3$  filter, where datapoints are becoming closer to their respective target states, but at a slow pace. This behavior suggests that as 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 unclear where the natural limit of the system is in this case, and a plateau could be reached at any moment.

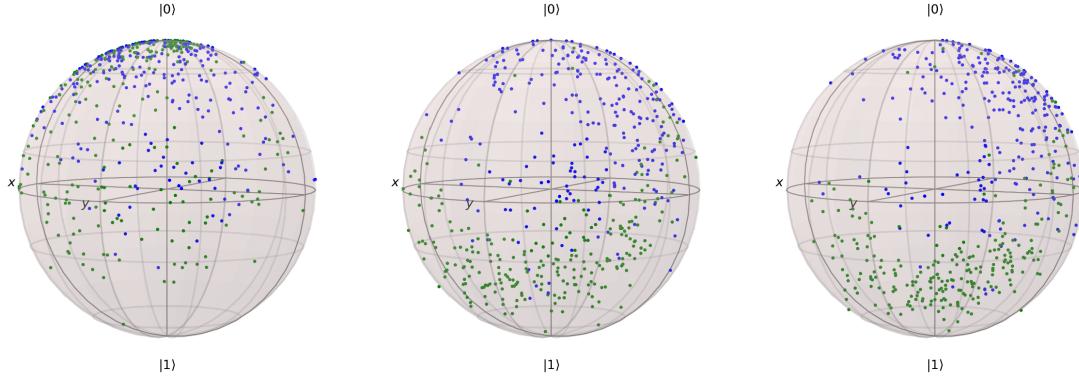


Fig. 6. Visualizations of Bloch sphere embeddings of datapoints corresponding to dataset images for the experiment conducted using a  $3 \times 3$  filter size. Left to right: Bloch spheres show train set data embeddings taken over epoch 1, epoch 2, and epoch 30. Point colors correspond to the images' respective class, where blue points represent class 0, and green points represent class 1.

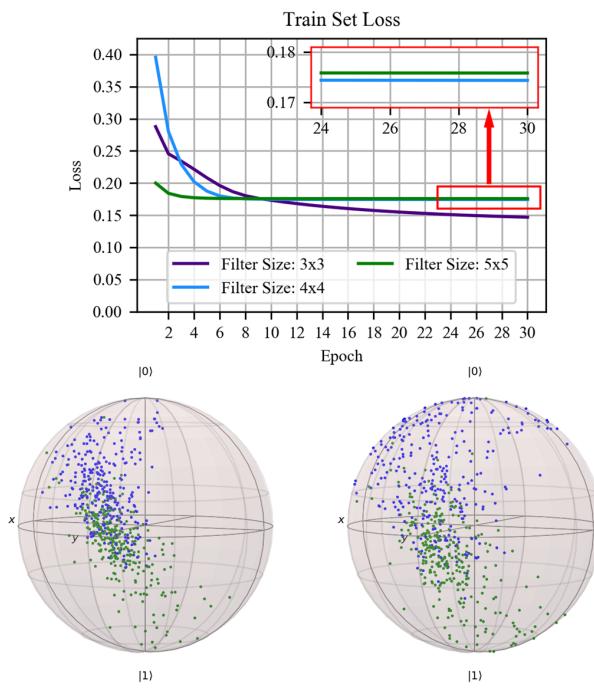


Fig. 7. Top: train set loss results of a facial identification task, using a single filter with varied size between experiments. The dataset consisted of AT&T face image data, combined with a selection of images taken from the CIFAR10 dataset. Bottom: Bloch sphere visualizations of train set image embeddings for the  $3 \times 3$  filter experiment. The left-hand sphere shows embeddings during epoch 10, whereas the right-hand sphere shows embeddings during epoch 30. Point colors correspond to the class of the embedded image, with blue for class 0 and green for class 1.

Regardless of any speculative analysis, the results achieved here are once again promising, and support the aims of this work 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 section. 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 on from the previous facial identification experimental results which contained a larger scale of data. Here, a training set of

TABLE V  
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	0.33594	0.500

seven images per class and a testing set of three images per class were used, with two classes of image in total.

For the results displayed in Table V, the classification accuracy for the training set of data was fairly poor for all experiments. While the testing set accuracy was fair for the experiment using a  $3 \times 3$  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.

This is again supported by result graphs shown in Fig. 8, 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  $3 \times 3$  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. While 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 individual's 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 preprocessing step was introduced; however, this is beyond

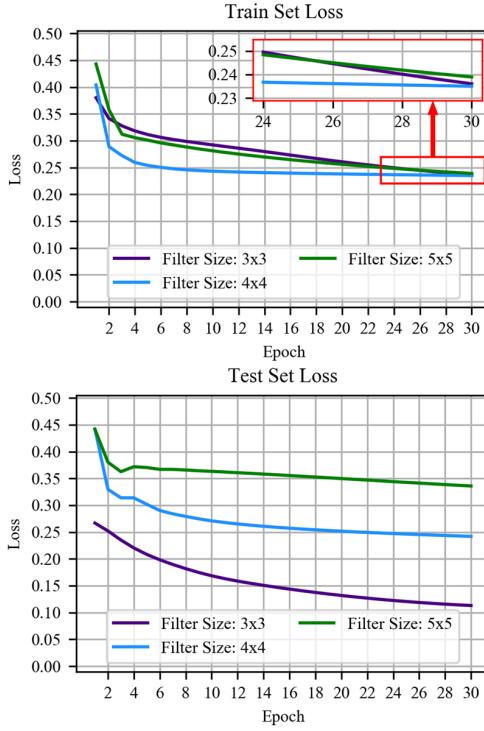


Fig. 8. Loss result curves for a binary face recognition task conducted using AT&T face image data.

the scope of this work and is a topic to be explored if the algorithm was specifically applied to a facial recognition task.

#### D. Environmental Noise Impact

In the current NISQ era of QC, 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 they 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 [35], [51].

Amplitude damping models energy relaxation within a qubit that occurs via interactions with the environment over time. More information on this can be found in [37] and [38]. Phase damping models environmental noise that affects the representation of quantum information, without changes being made to the status of excitation within the qubit itself. Phase damping can be modeled by the following Kraus operators,

where  $\lambda \in [0, 1]$  is the probability of qubit phase damping:

$$K_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{bmatrix}, K_1 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{bmatrix}. \quad (5)$$

The application of Kraus operator  $K_0$  does not affect the  $|0\rangle$  portion of the quantum state; however, it negatively impacts the  $|1\rangle$  portion by reducing its amplitude. This is the same operator that is used as 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 on the phase damping channel can also be found in [38].

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 the MNIST data, and so consistency between experiments was desired.

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

Charts displayed in Fig. 9 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. These charts display the evolution of both the train and test set loss and accuracy values as training epochs are conducted, up to a maximum of 30 epochs of optimization.

As can be seen throughout all curves of loss values, sharp plateaus occur very soon at the beginning of optimization, with considerably slower loss minimalization in general taking place after epoch 3. This is similar to accuracy charts as well, where any sharp improvements plateau at roughly epoch 3, before improving at a much slower rate.

The results displayed in Fig. 9 appear to follow the behavior 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 three 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

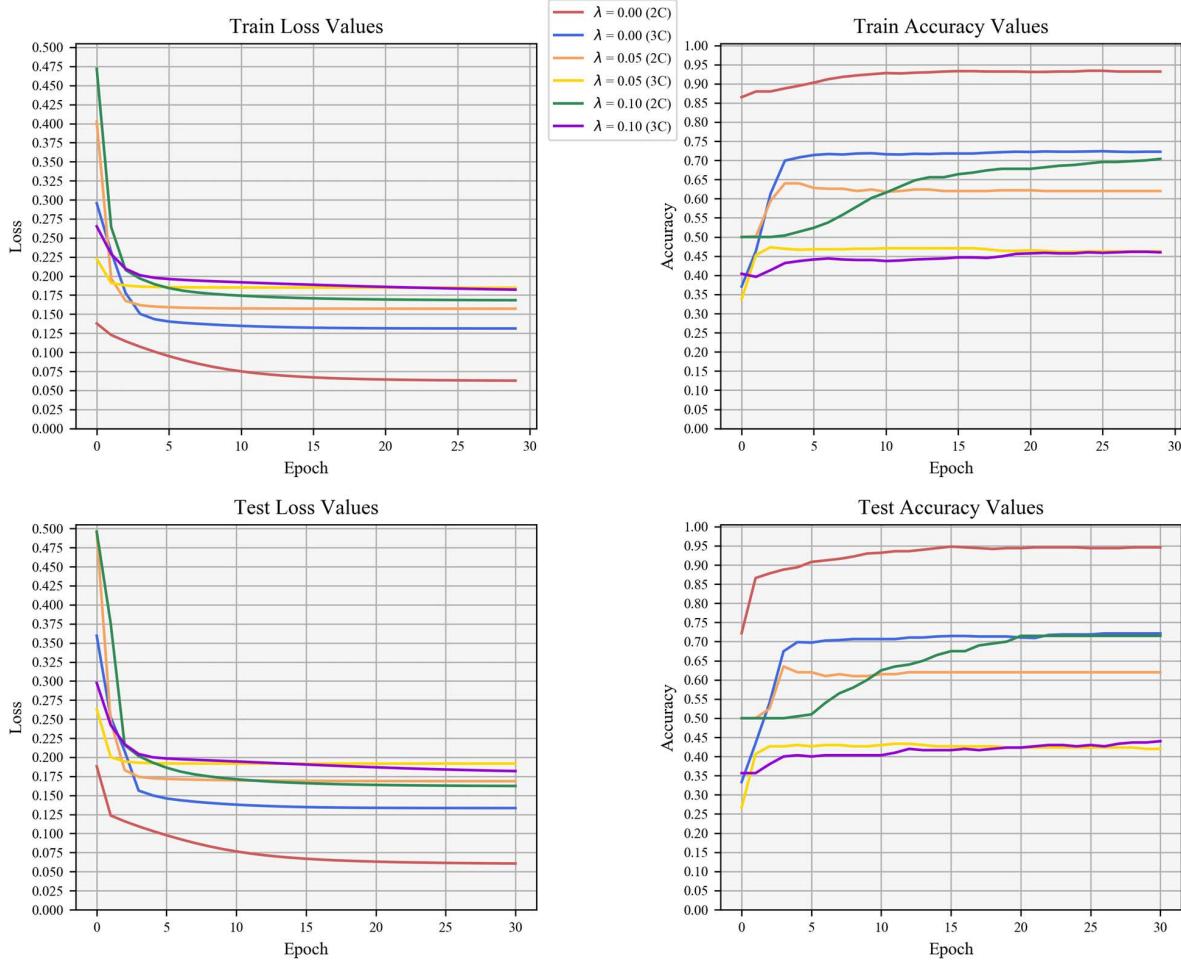


Fig. 9. Loss and classification accuracy values obtained through various binary (2C) and three-class (3C) experiments conducted in a noisy simulation environment using the amplitude damping channel. The experiments were conducted using a subset of the MNIST data (classes of 0, 1, and 2), a single filter of size  $3 \times 3$ , and  $\lambda$  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 I and Fig. 2.

decay via an amplitude damping channel. While initially this drop in performance is quite significant, the impact of noise with a greater magnitude is reduced.

Fig. 10 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. These charts display the evolution of both the train and test set loss and accuracy values as training epochs are conducted, up to a maximum of 30 epochs of optimization. Similar to those in Fig. 9, sharp improvements are seen for loss and accuracy values, which appear to plateau at approximately epoch 3, prior to learning at a considerably slower rate from then onward.

From the charts displayed in Fig. 10, the expected is followed for the most part. As the noise value  $\lambda$  increases, the loss values also increase for each task in turn. Similar to the behavior exhibited by the loss values, the final classification accuracy values also follow the behavior that would be expected somewhat.

Within this, the only experiment which does not follow this pattern is where  $\lambda = 0.05$  when using three 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 behavior 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, while 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.

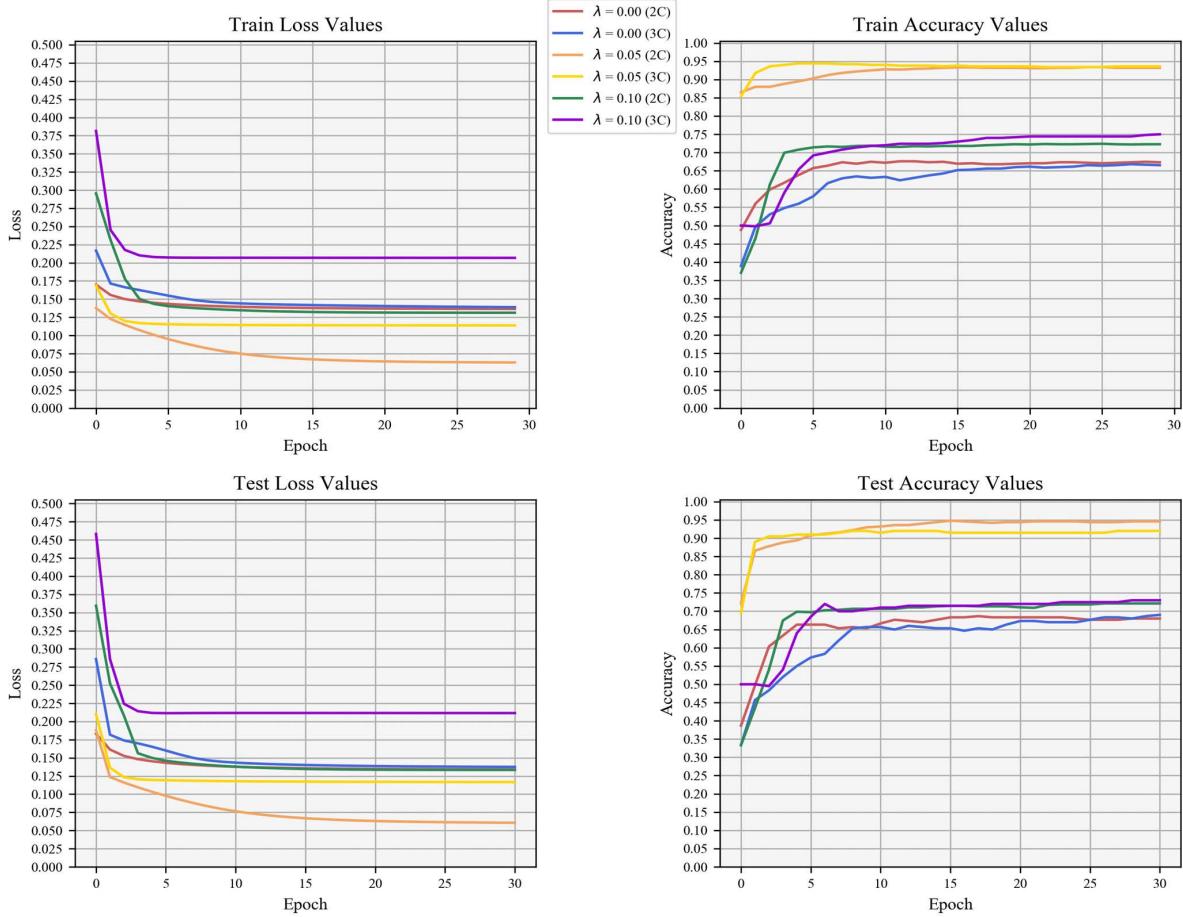


Fig. 10. Loss and classification accuracy values obtained through binary (2C) and three-class (3C) experiments conducted in a noisy simulation environment using the phase damping channel. These experiments were conducted using a subset of the MNIST data (classes of 0, 1, and 2), a single filter of size  $3 \times 3$ , and  $\lambda$  values ranging from 0.05 to 0.1. For clarity, the experiment with zero noise is a direct reference to the experimental result displayed in Table I and Fig. 2.

Regardless of whether the loss value increases, suggesting that 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 three-class task, where performance is comparable within approximately 10% and 5% for train set and test set, respectively, as  $\lambda$  was increased.

This is a promising factor to consider, as an innate robustness toward any kind of environmental noise can aid in optimization. In a case that the system was in a state capable of achieving more than 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.

## V. DISCUSSION

To summarize the findings discussed in Section IV in its entirety, initial experimental results have been displayed, in order to showcase early results 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 work 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 DL methodologies. However, the fact that the proposed method was able to reach test set accuracy scores in the 90% region within five epochs is promising in itself. While 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 three 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 more than 90% percentile. It is to be noted that only a single filter containing six parameters was implemented over the course of this work; 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. Here, the system showed promising results, reaching its highest classification accuracy values very close to

90% for a filter size of  $3 \times 3$ . In the context of this work, 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 they are considered promising and good in the context of the work and the experimental framework used. When applied to an additional task of facial recognition using AT&T 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 that the methodology was unable to learn and optimize effectively.

As the system was introduced to different environmental noise channels, initial results modeled using an amplitude damping channel suggest that noise greatly influences the qubit and reduces classification performance. However, when modeled 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.

While 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 [24]. With further development, it is hoped that any potential robustness can be 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 [52], and thus may provide some robustness to noisy environments with additional layers, in particular the amplitude channel [37]. Exploring modifications may aid in the robustness of the proposed method, and perhaps decrease any drop in classification as seen in Fig. 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 six parameters in total. As the field of DL has progressed from relatively shallow [20] to very deep networks consisting of many thousands of parameters [21], it should be considered here that the work being shown is proposed as a foundation or starting point to progress forward 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 work could be extended to provide additional insight and analysis into the feasibility of the algorithm as a quantum image classifier.

First, an aspect well noted throughout this work is that of the low number of filters, and subsequently parameters optimized in this implementation. While the point of this work 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. In addition, a usage of localized weights as described in Section III may also provide an advantage of maintaining spatial relationships between pixel values, without needing to increase the overall number 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 color 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 color 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 a lower number of classes are used; however, the ability to embed many datapoints into a very small section of the Bloch sphere will be difficult.

Therefore, naturally we will need 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 used is unknown. This point will also be undoubtedly affected by 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 [36] and [37].

It is noted that our new single-qubit CNN focuses on using as few qubits and parameters as possible. Here, we have proposed using a filter-based version of the existing method where the spatial relationship between data is preserved. In order to test a “maximally efficient” version of our quantum network, our experiments were carried out with a single convolutional filter that is applied everywhere on an image instead of many different filters that would each have their own parameters, where the loss function was modeled using the fidelity between the quantum state that is outputted by the unitaries and the pure quantum state that exists as a classification of an input.

In each experiment with three different types of problems (MNIST, fashion MNIST, and AT&T face database), our quantum network was able to create a filter size resulting in performances with a commendable quality. Although it is accepted that the leading CNN algorithms achieve a better performance accuracy, our goal was to test the method using a simple version of our QML having room for further improvement.

In our future work, we would like to extend our strategies to realize more complex architectures targeting higher performances, while the initial work in this article may serve as an important first step for what will be an exhaustive analysis of a specific type of QML algorithms.

## VI. CONCLUSION

In this work, a framework for efficient quantum image classification was proposed, using a minimum value of six parameters with 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 and phase damping channels, classification accuracy was reduced greatly 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.

Overall, the proposed methodology provides a solid foundation to progress forward to develop and build upon the success seen here using the bare minimum parameter and qubit count. As outlined in Section V, considerations for future work 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 toward modern-day image classification tasks that utilize high-resolution color 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.

#### ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

#### DATA AVAILABILITY STATEMENT

No new data was collected during this research project.

#### REFERENCES

- [1] L. Alzubaidi *et al.*, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, p. 53, Dec. 2021, doi: [10.1186/s40537-021-00444-8](https://doi.org/10.1186/s40537-021-00444-8).
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2017, doi: [10.1145/3065386](https://doi.org/10.1145/3065386).
- [3] G. Li, M. Zhang, J. Li, F. Lv, and G. Tong, “Efficient densely connected convolutional neural networks,” *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107610, doi: [10.1016/j.patcog.2020.107610](https://doi.org/10.1016/j.patcog.2020.107610).
- [4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
- [7] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
- [8] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, *arXiv:1804.02767*.
- [9] H. A. Haenssle *et al.*, “Man against machine: Diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists,” *Ann. Oncol.*, vol. 29, no. 8, pp. 1836–1842, Aug. 2018, doi: [10.1093/ANNONC/MDY166](https://doi.org/10.1093/annonc/mdy166).
- [10] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, Jan. 2017, doi: [10.1038/nature21056](https://doi.org/10.1038/nature21056).
- [11] M. Cicero *et al.*, “Training and validating a deep convolutional neural network for computer-aided detection and classification of abnormalities on frontal chest radiographs,” *Invest. Radiol.*, vol. 52, no. 5, pp. 281–287, May 2017, doi: [10.1097/RLI.0000000000000341](https://doi.org/10.1097/RLI.0000000000000341).
- [12] J.-Z. Cheng *et al.*, “Computer-aided diagnosis with deep learning architecture: Applications to breast lesions in U.S. Images and pulmonary nodules in CT scans,” *Sci. Rep.*, vol. 6, no. 1, pp. 1–13, Apr. 2016, doi: [10.1038/srep24454](https://doi.org/10.1038/srep24454).
- [13] B. Wang, D. Zhang, D. Zhang, P. J. Brantingham, and A. L. Bertozzi, “Deep learning for real time crime forecasting,” 2017, *arXiv:1707.03340*.
- [14] S. Yadav, “Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 8, no. 7, pp. 1368–1375, Jul. 2020, doi: [10.22214/ijraset.2020.30560](https://doi.org/10.22214/ijraset.2020.30560).
- [15] M. Wu, W. Xie, X. Shi, P. Shao, and Z. Shi, “Real-time drone detection using deep learning approach,” *Machine Learning and Intelligent Communications* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering) vol. 251, 2018, pp. 22–32, doi: [10.1007/978-3-030-00557-3\\_3](https://doi.org/10.1007/978-3-030-00557-3_3).
- [16] I. Psychoula *et al.*, “A deep learning approach for privacy preservation in assisted living,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 710–715, doi: [10.1109/PERCOMW.2018.8480247](https://doi.org/10.1109/PERCOMW.2018.8480247).
- [17] E. Merdivan. (2019). *Ambient Assisted Living with Deep Learning*. Accessed: Sep. 16, 2021. [Online]. Available: <https://tel.archives-ouvertes.fr/tel-02927785/>
- [18] M. A. Qureshi, K. N. Qureshi, G. Jeon, and F. Piccialli, “Deep learning-based ambient assisted living for self-management of cardiovascular conditions,” *Neural Comput. Appl.*, vol. 33, pp. 1–19, Jan. 2021, doi: [10.1007/s00521-020-05678-w](https://doi.org/10.1007/s00521-020-05678-w).
- [19] U. Malūkas, R. Maskeliūnas, R. Damaševičius, and M. Woźniak, “Real time path finding for assisted living using deep learning,” *J. Universal Comput. Sci.*, vol. 24, no. 4, pp. 475–487, Apr. 2018.
- [20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [21] Y. Assiri, “Stochastic optimization of plain convolutional neural networks with simple methods,” 2020, *arXiv:2001.08856*.
- [22] E. Farhi and H. Neven, “Classification with quantum neural networks on near term processors,” 2018, *arXiv:1802.06002*.
- [23] C. M. Wilson *et al.*, “Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers,” 2018, *arXiv:1806.08321*.
- [24] W. Cappelletti, R. Erbanni, and J. Keller, “Polyadic quantum classifier,” 2020, *arXiv:2007.14044*.
- [25] J. Bausch, “Recurrent quantum neural networks,” in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, Dec. 2020.
- [26] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, “Variational quantum circuits for deep reinforcement learning,” 2019, *arXiv:1907.00397*.
- [27] J.-M. Liang, S.-Q. Shen, M. Li, and L. Li, “Variational quantum algorithms for dimensionality reduction and classification,” *Phys. Rev. A, Gen. Phys.*, vol. 101, no. 3, Mar. 2020, Art. no. 032323, doi: [10.1103/PhysRevA.101.032323](https://doi.org/10.1103/PhysRevA.101.032323).
- [28] V. Havlíček *et al.*, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: [10.1038/s41586-019-0980-2](https://doi.org/10.1038/s41586-019-0980-2).
- [29] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, “Transfer learning in hybrid classical-quantum neural networks,” 2019, *arXiv:1912.08278*.
- [30] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quanvolutional neural networks: Powering image recognition with quantum circuits,” *Quantum Mach. Intell.*, vol. 2, no. 1, pp. 1–9, Jun. 2020, doi: [10.1007/s42484-020-00012-y](https://doi.org/10.1007/s42484-020-00012-y).
- [31] I. Cong, S. Choi, and M. D. Lukin, “Quantum convolutional neural networks,” *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, Dec. 2019, doi: [10.1038/s41567-019-0648-8](https://doi.org/10.1038/s41567-019-0648-8).
- [32] I. Kerenidis, J. Landman, and A. Prakash, “Quantum algorithms for deep convolutional neural networks,” 2019, *arXiv:1911.01117*.
- [33] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, “Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms,” *Adv. Quantum Technol.*, vol. 2, no. 12, Dec. 2019, Art. no. 1900070, doi: [10.1002/qute.201900070](https://doi.org/10.1002/qute.201900070).
- [34] T. Hubregtsen, P. Josef, J. Pichlmeier, P. Stecher, and K. Bertels, *Evaluation of Parameterized Quantum Circuits: On the Relation Between Classification Accuracy, Expressibility and Entangling Capability*. Accessed: Sep. 28, 2020. [Online]. Available: <https://www.researchgate.net/publication/340115185>
- [35] R. LaRose and B. Coyle, “Robust data encodings for quantum classifiers,” 2020, *arXiv:2003.01695*.

- [36] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” *Quantum*, vol. 4, p. 226, Feb. 2020, doi: [10.22331/q-2020-02-06-226](https://doi.org/10.22331/q-2020-02-06-226).
- [37] P. Easom-Mccaldin, A. Bouridane, A. Belatreche, and R. Jiang, “On depth, robustness and performance using the data re-uploading single-qubit classifier,” *IEEE Access*, vol. 9, pp. 65127–65139, 2021, doi: [10.1109/ACCESS.2021.3075492](https://doi.org/10.1109/ACCESS.2021.3075492).
- [38] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, vol. 27, no. 3. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [39] L. Deng, “The MNIST database of handwritten digit images for machine learning research,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012, doi: [10.1109/MSP.2012.2211477](https://doi.org/10.1109/MSP.2012.2211477).
- [40] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017, *arXiv:1708.07747*.
- [41] F. S. Samaria and A. C. Harter, “Parameterisation of a stochastic model for human face identification,” in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142, doi: [10.1109/ACV.1994.341300](https://doi.org/10.1109/ACV.1994.341300).
- [42] A. Krizhevsky and G. Hinton. (2009). *Learning Multiple Layers of Features From Tiny Images*. Accessed: Sep. 16, 2021. [Online]. Available: <http://www.cs.toronto.edu/~kriz/cifar.html>
- [43] V. Bergholm *et al.*, “PennyLane: Automatic differentiation of hybrid quantum-classical computations,” 2018, *arXiv:1811.04968*.
- [44] A. Paszke *et al.* (2017). *Automatic Differentiation in Pytorch*. Accessed: Sep. 16, 2021. [Online]. Available: <https://openreview.net/forum?id=BJJsrmfCZ>
- [45] Y. Suzuki *et al.*, “Qulacs: A fast and versatile quantum circuit simulator for research purpose,” 2020, *arXiv:2011.13524*.
- [46] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, “Barren plateaus in quantum neural network training landscapes,” *Nature Commun.*, vol. 9, no. 1, pp. 1–6, Dec. 2018, doi: [10.1038/s41467-018-07090-4](https://doi.org/10.1038/s41467-018-07090-4).
- [47] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, “Cost function dependent barren plateaus in shallow parametrized quantum circuits,” *Nature Commun.*, vol. 12, no. 1, pp. 1–12, Dec. 2021, doi: [10.1038/s41467-021-21728-w](https://doi.org/10.1038/s41467-021-21728-w).
- [48] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, “Quantum natural gradient,” *Quantum*, vol. 4, p. 269, May 2020, doi: [10.22331/Q-2020-05-25-269](https://doi.org/10.22331/Q-2020-05-25-269).
- [49] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, “An initialization strategy for addressing barren plateaus in parametrized quantum circuits,” *Quantum*, vol. 3, p. 214, Dec. 2019, doi: [10.22331/q-2019-12-09-214](https://doi.org/10.22331/q-2019-12-09-214).
- [50] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Dec. 2015, pp. 1–15.
- [51] C. Ciliberto *et al.*, “Quantum machine learning: A classical perspective,” *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 474, no. 2209, Jan. 2018, Art. no. 20170551, doi: [10.1098/rspa.2017.0551](https://doi.org/10.1098/rspa.2017.0551).
- [52] M. Schuld, R. Swoke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Phys. Rev. A, Gen. Phys.*, vol. 103, no. 3, Mar. 2021, Art. no. 032430, doi: [10.1103/PhysRevA.103.032430](https://doi.org/10.1103/PhysRevA.103.032430).
- [53] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Phys. Rev. Lett.*, vol. 113, no. 13, Sep. 2014, Art. no. 130503, doi: [10.1103/PhysRevLett.113.130503](https://doi.org/10.1103/PhysRevLett.113.130503).
- [54] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017, doi: [10.1038/nature23474](https://doi.org/10.1038/nature23474).
- [55] N. Wiebe, D. Braun, and S. Lloyd, “Quantum algorithm for data fitting,” *Phys. Rev. Lett.*, vol. 109, no. 5, Aug. 2012, Art. no. 050505, doi: [10.1103/PhysRevLett.109.050505](https://doi.org/10.1103/PhysRevLett.109.050505).



**Philip Easom-McCaldin** received the B.Sc. degree (Hons.) in computer science and the Ph.D. degree in computer science and quantum machine learning, under the supervision of Prof. Ahmed Bouridane, from Northumbria University, Newcastle upon Tyne, U.K., in 2018 and 2022, respectively.

He is currently with the Computer and Information Sciences Department, Northumbria University, Newcastle upon Tyne U.K. His research interests include machine learning, quantum computing, and their combined applications.



**Ahmed Bouridane** (Senior Member, IEEE) received the “Ingenieur d’Etat” degree in electronics from the École Nationale Polytechnique of Algiers (ENPA), El Harrach, Algeria, in 1982, the M.Phil. degree in electrical engineering (VLSI design for signal processing) from the University of Newcastle upon Tyne, Newcastle upon Tyne, U.K., in 1988, and the Ph.D. degree in electrical engineering (computer vision) from the University of Nottingham, Nottingham, U.K., in 1992.

From 1992 to 1994, he was a Research Developer in telesurveillance and access control applications. In 1994, he joined Queen’s University Belfast, Belfast, U.K., initially as a Lecturer in computer architecture and image processing and later on he was promoted to a Reader in computer science. He is currently a Full Professor with the Department of Computer Engineering, University of Sharjah, Sharjah, United Arab Emirates. He has authored or coauthored more than 350 publications and one research book on imaging for forensics and security. His research interests are imaging for forensics and security, biometrics, homeland security, image/video watermarking, medical engineering, cryptography, and mobile and visual computing.



**Ammar Belatreche** (Member, IEEE) received the Ph.D. degree in computer science from Ulster University, Derry, U.K., in 2006.

He joined Northumbria University, Newcastle upon Tyne, U.K., in May 2016. He was a Research Associate with the Intelligent Systems Research Centre (ISRC) and then a Lecturer in computer science with the School of Computing and Intelligent Systems, Ulster University. He is currently an Associate Professor in computer science and the Programme Leader of the M.Sc. advanced computer science with the Department of Computer and Information Sciences. He is also a member of the Intelligent Systems Research Group (ISRG). He has extensive experience across academic and research and development in the areas of machine learning and AI, bio-inspired intelligent systems, structured and unstructured data analytics, capital markets engineering, and image processing and understanding. He has led a number of research and consultancy projects and has successfully supervised/co-supervised ten Ph.D. students to completion.

Dr. Belatreche is a fellow of the Higher Education Academy. He has served as a Program Committee Member and a Reviewer for several international conferences and journals. He is an Associate Editor of *Neurocomputing* (Elsevier).



**Richard Jiang** is currently a Senior Lecturer (an Associate Professor) with the School of Computing and Communications, Lancaster University, Lancaster, U.K. He is also an Associate Member of the EPSRC College and an EPSRC RISE Connector. He has supervised or co-supervised more than ten Ph.D. students. He has authored more than 80 publications. His research interest mainly resides in the fields of artificial intelligence, explainable artificial intelligence (XAI), neural computation, biomedical image analysis, intelligent systems, and biometrics and privacy. His current research has been supported by grants from EPSRC, Leverhulme Trust, Qatar Science Foundation, and other industry funders.

Mr. Jiang is a fellow of Advance Higher Education (HEA). He was a Lead Editor of two Springer books.

**Somaya Al-Maadeed** (Senior Member, IEEE) received the Ph.D. degree in computer science from University of Nottingham, U.K., in 2004.

She is currently with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar, and the Coordinator of the Computer Vision and AI Research Group. She enjoys excellent collaboration with national and international institutions and industry. She is a Principal Investigator of several funded research projects generating approximately five million. She has published extensively in the field of pattern recognition and delivered workshops on teaching programming for undergraduate students. She has attended workshops related to higher education strategy, assessment methods, and interactive teaching.

Dr. Al-Maadeed was elected as the IEEE Chair for the Qatar Section in 2015.